

## LAMPIRAN

### Hasil Kuesioner

#### a. Siswa

Pertanyaan	Hasil
<b>1. Masukkan Kelas Anda</b>	
Kelas X	2
Kelas XI	14
Kelas XII	97
<b>2. Masukkan umur Anda</b>	
15	6
16	11
17	55
18	41
<b>3. Apakah Anda menyimpan dokumen try out masuk perguruan tinggi</b>	
Ya	51
Tidak	62
<b>4. Apa yang Anda simpan</b>	
Soal	51
Hasil try out (Jawaban soal)	17
Nilai try out	12
<b>5. Untuk apa Anda menyimpannya</b>	
Belajar	31
Tolak ukur kemampuan setiap mata pelajaran	15
Perbandingan nilai try out	5
<b>6. Bagaimana cara Anda menyimpannya</b>	
Dimasukkan menjadi satu (stopmap)	32
Dipisahkan per try out	6
Dipisahkan per pelajaran	13
<b>7. Apakah Anda merasa kesulitan menyimpan dokumen- dokumen tersebut</b>	
Ya	7
Tidak	44
<b>8. Kesulitan apa yang Anda hadapi saat menyimpan dokumen- dokumen try out</b>	
Dokumen hilang	1

Kurang rapi dalam menyimpan dokumen dari sekolah atau bimbel	3
Tidak menerima dokumen dari sekolah atau bimbel	1
Malas dalam menyimpan dokumen	2
9. Menurut Anda, sebenarnya apakah penting untuk menyimpan dokumen-dokumen try out masuk perguruan tinggi?	
Ya	56
Tidak	6
10. Mengapa hal tersebut penting	
Belajar	14
Tolak ukur kemampuan setiap mata pelajaran	37
Perbandingan nilai try out	4
Karena segala dokumen berharga penting untuk disimpan. Sebab takut diperlukan dikemudian hari	1
11. Mengapa hal tersebut tidak penting ? karena	
Malas	2
Ribet	1
Data tidak lengkap	2
Data di komputer	0
Soal tidak sama dengan yang diujikan	1
12. Apakah Anda mengetahui di mata pelajaran apa kekurangan dan kelebihan Anda	
Ya	108
Tidak	5
13. Di mata pelajaran apa kekurangan Anda	
Matematika	39
Fisika	29
Kimia	13
Biologi	4
Sosiologi	0
Ekonomi	2
Sejarah	5
Geografi	2
Bahasa Indonesia	1
Bahasa Inggris	13
14. Seberapa yakinkah bahwa Anda lemah di mata pelajaran tersebut	

Sangat tidak yakin	2
Tidak yakin	11
Kurang yakin	46
Yakin	37
Sangat yakin	12
<b>15. Bagaimana proses mengetahui kekurangan Anda</b>	
Melihat nilai- nilai	22
Semangat belajar	15
Kesulitan memahami pelajaran	63
Dari guru pengajar disekolah atau bimbil	3
Kebanyakan rumus	4
Saya dari ipa	1
<b>16. Di mata pelajaran apa kelebihan Anda</b>	
Matematika	24
Fisika	5
Kimia	10
Biologi	21
Sosiologi	3
Ekonomi	9
Sejarah	4
Geografi	3
Bahasa Indonesia	16
Bahasa Inggris	13
<b>17. Seberapa yakinkah bahwa Anda lebih di mata pelajaran tersebut</b>	
Sangat tidak yakin	4
Tidak yakin	11
Kurang yakin	12
Yakin	57
Sangat yakin	22
<b>18. Bagaimana proses mengetahui kelebihan Anda</b>	
Melihat nilai- nilai	13
Semangat belajar	33
Kemudahan memahami pelajaran	57
Dari guru pengajar di sekolah atau bimbil	5
<b>19. Apakah anda membutuhkan aplikasi yang dapat menampilkan pemberitahuan kelebihan dan kekurangan mata pelajaran</b>	
Sangat tidak yakin	5

Tidak yakin	5
Kurang yakin	21
Yakin	41
Sangat yakin	41
20. Apakah Anda membutuhkan aplikasi yang dapat merekap hasil ujian secara otomatis	
Sangat tidak yakin	3
Tidak yakin	2
Kurang yakin	15
Yakin	49
Sangat yakin	44
21. Apakah Anda membutuhkan aplikasi yang dapat menampilkan progress atau perkembangan hasil ujian dalam bentuk grafik/ visual	
Sangat tidak yakin	2
Tidak yakin	4
Kurang yakin	16
Yakin	50
Sangat yakin	41
22. Apakah Anda membutuhkan aplikasi yang menampilkan pembahasan soal ujian yang sudah dikerjakan	
Sangat tidak yakin	2
Tidak yakin	0
Kurang yakin	16
Yakin	40
Sangat yakin	55
23. Apakah Anda membutuhkan aplikasi yang dapat menampilkan soal- soal yang menjadi kelebihan Anda	
Sangat tidak yakin	2
Tidak yakin	1
Kurang yakin	15
Yakin	48
Sangat yakin	47
24. Apakah Anda membutuhkan aplikasi yang dapat menampilkan soal- soal yang menjadi kekurangan Anda	
Sangat tidak yakin	2
Tidak yakin	0
Kurang yakin	26
Yakin	41
Sangat yakin	44

## b. Mahasiswa Baru

Pertanyaan	Hasil
<b>1. Masukkan Umur Anda</b>	
17	9
18	65
19	22
20	6
<b>2. Apakah Anda menyimpan dokumen try out masuk perguruan tinggi</b>	
Ya	78
Tidak	24
<b>3. Apa yang Anda simpan</b>	
Soal	74
Hasil try out (Jawaban soal)	11
Nilai try out	28
<b>4. Untuk apa Anda menyimpannya</b>	
Belajar	55
Tolak ukur kemampuan setiap mata pelajaran	19
Perbandingan nilai try out	3
Kenang- kenangan aja	1
<b>5. Bagaimana cara Anda menyimpannya</b>	
Dimasukkan menjadi satu (stopmap)	49
Dipisahkan per try out	24
Dipisahkan per pelajaran	5
<b>6. Apakah Anda merasa kesulitan menyimpan dokumen- dokumen tersebut</b>	
Ya	49
Tidak	29
<b>7. Kesulitan apa yang Anda hadapi saat menyimpan dokumen- dokumen try out</b>	
Dokumen hilang	12
Kurang rapi dalam menyimpan dokumen dari sekolah atau bimbel	35
Tidak menerima dokumen dari sekolah atau bimbel	1
Malas dalam menyimpan dokumen	1
<b>8. Menurut Anda, sebenarnya apakah penting untuk menyimpan dokumen- dokumen try out masuk perguruan tinggi?</b>	
Ya	13

Tidak	11
<b>9. Mengapa hal tersebut penting</b>	
Belajar	3
Tolak ukur kemampuan setiap mata pelajaran	8
Perbandingan nilai try out	0
<b>10. Mengapa hal tersebut tidak penting ? karena</b>	
Malas	1
Ribet	6
Data tidak lengkap	0
Data di komputer	4
Setiap tahunnya beda- beda	1
Tidak ada gunanya	1
<b>11. Apakah Anda mengetahui di mata pelajaran apa kekurangan dan kelebihan Anda</b>	
Ya	100
Tidak	2
<b>12. Di mata pelajaran apa kekurangan Anda</b>	
Matematika	50
Fisika	22
Kimia	8
Biologi	1
Sosiologi	1
Ekonomi	3
Sejarah	2
Geografi	1
Bahasa Indonesia	4
Bahasa Inggris	8
<b>13. Seberapa yakinkah bahwa Anda lemah di mata pelajaran tersebut</b>	
Sangat tidak yakin	1
Tidak yakin	1
Kurang yakin	26
Yakin	52
Sangat yakin	20
<b>14. Bagaimana proses mengetahui kekurangan Anda</b>	
Melihat nilai- nilai	33
Semangat belajar	11
Kesulitan memahami pelajaran	54

Dari guru pengajar di sekolah atau bimbel	2
<b>15. Di mata pelajaran apa kelebihan Anda</b>	
Matematika	17
Fisika	2
Kimia	11
Biologi	28
Sosiologi	6
Ekonomi	5
Sejarah	3
Geografi	3
Bahasa Indonesia	10
Bahasa Inggris	15
<b>16. Seberapa yakinkah bahwa Anda lebih di mata pelajaran tersebut</b>	
Sangat tidak yakin	0
Tidak yakin	1
Kurang yakin	19
Yakin	60
Sangat yakin	19
<b>17. Bagaimana proses mengetahui kelebihan Anda</b>	
Melihat nilai- nilai	30
Semangat belajar	17
Kemudahan memahami pelajaran	49
Dari guru pengajar di sekolah atau bimbel	3
Pendalaman materi sendiri	1
<b>18. Apakah anda membutuhkan aplikasi yang dapat menampilkan pemberitahuan kelebihan dan kekurangan mata pelajaran</b>	
Sangat tidak yakin	2
Tidak yakin	1
Kurang yakin	7
Yakin	54
Sangat yakin	38
<b>19. Apakah Anda membutuhkan aplikasi yang dapat merekap hasil ujian secara otomatis</b>	
Sangat tidak yakin	0
Tidak yakin	0
Kurang yakin	6
Yakin	46

Sangat yakin	50
20. Apakah Anda membutuhkan aplikasi yang dapat menampilkan progress atau perkembangan hasil ujian dalam bentuk grafik/ visual	
Sangat tidak yakin	1
Tidak yakin	0
Kurang yakin	5
Yakin	45
Sangat yakin	51
21. Apakah anda membutuhkan aplikasi yang menampilkan pembahasan soal ujian yang sudah dikerjakan	
Sangat tidak yakin	1
Tidak yakin	0
Kurang yakin	3
Yakin	42
Sangat yakin	56
22. Apakah anda membutuhkan aplikasi yang dapat menampilkan soal- soal yang menjadi kelebihan Anda	
Sangat tidak yakin	1
Tidak yakin	1
Kurang yakin	4
Yakin	42
Sangat yakin	54
23. Apakah Anda membutuhkan aplikasi yang dapat menampilkan soal- soal yang menjadi kekurangan Anda	
Sangat tidak yakin	1
Tidak yakin	0
Kurang yakin	5
Yakin	41
Sangat yakin	55



## Source Code Controller Sistem

### a. UjianController.cs

```

using System;
using System.Collections.Generic;
using System.Data;
using System.Data.Entity;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.Mvc;
using Microsoft.AspNet.Identity;
using SkripsiTagBasedExam.Models;
using Microsoft.VisualStudio.TestTools.UnitTesting;
using System.Security.Principal;

namespace SkripsiTagBasedExam.Controllers
{
    public class UjianController : Controller
    {
        private examdbEntities db = new examdbEntities();
        [Authorize(Roles = "Guru")]
        public ActionResult Index()
        {
            try
            {
                string id_guru = User.Identity.GetUserId();
                DateTime today = DateTime.Now;

                UjianData.Status.UjianData data = new UjianData.Status.UjianData();
                data.done = db.Ujian.Include(u => u.DataGuru).Include(u => u.MataPelajaran).Where(u => u.Mulai_Ujian < today && u.Createdby == id_guru).ToList();
            }
        }
    }
}

```

```

        data.undone = db.Ujian.Include(u => u.DataGuru).Include(u => u.MataPelajaran).Where(u => u.Mulai_Ujian > today && u.Createdby == id_guru).ToList();
        //var ujians = db.Ujians.Include(u => u.DataGuru).Include(u => u.MataPelajaran);
        ViewBag.Status = "SUCCESS";
        ViewData["Guru"] = db.DataGuru.Find(id_guru); // TO BE CHANGED
        return View(data);
    }
    catch
    {
        ViewBag.Status = "FAIL";
        return View();
    }
}

protected override void Dispose(bool disposing)
{
    if (disposing)
    {
        db.Dispose();
    }
    base.Dispose(disposing);
}

[HttpPost]
public int DeleteUjian(int id_ujian)
{
    string id_guru = User.Identity.GetUserId();
    int i = 0;
    try
    {

```

```

        string res =
DeleteUjianTest(id_guru, id_ujian);

        i = 1;
    }
    catch
    {
        i = 0;
    }

    return i;
}
[HttpPost]
[Authorize(Roles = "Guru")]
public ActionResult
UpdateUjian(int id_ujian, string
Nama_Ujian, string Enrolment_Key,
DateTime Mulai_Ujian, DateTime
Batas_Ujian, int Durasi_pengerjaan)
{
    string id_guru =
User.Identity.GetUserId();
    try
    {
        string res =
UpdateUjianTest(id_guru, id_ujian,
Nama_Ujian, Enrolment_Key,
Mulai_Ujian, Batas_Ujian,
Durasi_pengerjaan);
        return
RedirectToAction("Index",
"DetailUjian", new { id =
db.Ujian.Find(id_ujian).Ujian_Id });
    }
    catch
    {
        return
RedirectToAction("Index");
    }
}
[HttpPost]
[Authorize(Roles = "Guru")]
public ActionResult
InsertUjian(string Nama_Ujian, string
Enrolment_Key, DateTime Mulai_Ujian,

```

```

DateTime Batas_Ujian, int
Durasi_pengerjaan)
{
    string id_guru =
User.Identity.GetUserId();
    string i =
InsertUjianTest(id_guru, Nama_Ujian,
Enrolment_Key, Mulai_Ujian,
Batas_Ujian, Durasi_pengerjaan);
    return
RedirectToAction("Index");
}

//TEST
public string
InsertUjianTest(string id_guru, string
Nama_Ujian, string Enrolment_Key,
DateTime Mulai_Ujian, DateTime
Batas_Ujian, int Durasi_pengerjaan)
{
    try
    {
        Ujian ujian = new
Ujian();
        ujian>Nama_Ujian =
Nama_Ujian;
        ujian.Createdby =
id_guru;
        ujian.MP_Id =
db.DataGuru.Find(id_guru).MP_Id;
        ujian.EnrolmentKey =
Enrolment_Key;
        ujian.Tanggal_Upload =
DateTime.Now;
        ujian.Mulai_Ujian =
Mulai_Ujian;
        ujian.Batas_ujian =
Batas_Ujian;

        ujian.Durasi_pengerjaan =
Durasi_pengerjaan;
        db.Ujian.Add(ujian);
        db.SaveChanges();
        return "SUCCESS";
    }
}

```

```

    }
    catch(Exception ex)
    {
        return ex.Message;
    }
}

public string
UpdateUjianTest(string id_guru, int
id_ujian, string Nama_Ujian, string
Enrolment_Key, DateTime Mulai_Ujian,
DateTime Batas_Ujian, int
Durasi_pengerjaan)
{
    try
    {
        Ujian ujian =
db.Ujian.Find(id_ujian);
        if (ujian.Createdby ==
id_guru)
        {
            ujian.Nama_Ujian =
Nama_Ujian;
            ujian.EnrolmentKey
= Enrolment_Key;
            ujian.Mulai_Ujian
= Mulai_Ujian;
            ujian.Batas_ujian
= Batas_Ujian;

            ujian.Durasi_pengerjaan =
Durasi_pengerjaan;

            db.Entry(ujian).State =
EntityState.Modified;
            db.SaveChanges();

            return "SUCCESS";
        }
        else
        {
            return
"INVALID_CREATOR";
        }
    }
}

```

```

    }
    catch(Exception ex)
    {
        return ex.Message;
    }
}

public string
DeleteUjianTest(string id_guru, int
id_ujian)
{
    try
    {
        Ujian ujian =
db.Ujian.Find(id_ujian);

        if (ujian.Createdby ==
id_guru)
        {
            db.Ujian.Remove(ujian);

            db.SaveChanges();
            return "SUCCESS";
        }
        else
        {
            return
"INVALID_CREATOR";
        }
    }
    catch(Exception ex)
    {
        return ex.Message;
    }
}

namespace UjianData.Status
{
    public class UjianData
    {

```

```

        public List<Ujian> undone {
get; set; }
        public List<Ujian> done { get;
set; }
        public List<Ujian> ongoing {
get; set; }
        public string Status { get;
set; }

        public UjianData()
        {
            undone = new List<Ujian>();
            done = new List<Ujian>();
            ongoing = new
List<Ujian>();
            Status = "NOT SUCCESS";
        }
    }
}

namespace UjianData.Tests
{
    public class
TestableControllerContext
: ControllerContext
    {
        public TestableHttpContext
TestableHttpContext { get; set; }
    }

    public class TestableHttpContext :
HttpContextBase
    {
        public override IPrincipal User
{ get; set; }
    }

    [TestClass]
    public class UjianTests
    {
        GenericIdentity identity = new
GenericIdentity("Guru");

SkripsiTagBasedExam.Controllers.UjianC
ontroller controller = new

```

```

SkripsiTagBasedExam.Controllers.UjianC
ontroller();

        int id_ujian = 21;

        [TestMethod]
        public void IndexTest()
        {
            var controllerContext = new
TestableControllerContext();
            var principal = new
GenericPrincipal(identity, null);
            var testableHttpContext =
new TestableHttpContext
            {
                User = principal
            };

            controllerContext.HttpContext
=
testableHttpContext;

            controller.ControllerContext
=
controllerContext;

            var result =
controller.Index() as ViewResult;

            Assert.AreEqual("SUCCESS",
result.ViewBag.Status);
        }

        [TestMethod]
        public void InsertUjian()
        {
            string Nama_Ujian = "TEST["
+
DateTime.Now.ToString("yyyyMMddHHmmss"
) + "];"
            string Enrolment_Key =
"HelloTest";
            DateTime Mulai_Ujian =
DateTime.Parse("12/12/2017 13:00");

```

```

        DateTime Batas_Ujian =
DateTime.Parse("12/12/2017 16:00");
        int Durasi_pengerjaan =
180;
        string id_guru = "36b83bf0-
1330-4b49-beda-2673249e625d";

        string res =
controller.InsertUjianTest(id_guru,
Nama_Ujian, Enrolment_Key,
Mulai_Ujian, Batas_Ujian,
Durasi_pengerjaan);

        Assert.AreEqual("SUCCESS",
res);
    }

    [TestMethod]
    public void UpdateUjianA()
    {
        string Nama_Ujian = "TEST["
+
DateTime.Now.ToString("yyyyMMddHHmss"
) + "];
        string Enrolment_Key =
"Hello12";
        DateTime Mulai_Ujian =
DateTime.Parse("12/12/2017 13:00");
        DateTime Batas_Ujian =
DateTime.Parse("12/12/2017 16:00");
        int Durasi_pengerjaan =
180;
        string id_guru = "61210bfb-
d1c3-41b2-8fe4-4c11033ab97a";

        string res =
controller.UpdateUjianTest(id_guru,
id_ujian, Nama_Ujian, Enrolment_Key,
Mulai_Ujian, Batas_Ujian,
Durasi_pengerjaan);

        Assert.AreEqual("INVALID_CREATOR",
res);
    }

```

```

    }

    [TestMethod]
    public void UpdateUjianB()
    {
        string Nama_Ujian = "TEST["
+
DateTime.Now.ToString("yyyyMMddHHmss"
) + "];
        string Enrolment_Key =
"Hello34";
        DateTime Mulai_Ujian =
DateTime.Parse("12/12/2017 13:00");
        DateTime Batas_Ujian =
DateTime.Parse("12/12/2017 16:00");
        int Durasi_pengerjaan =
180;
        string id_guru = "36b83bf0-
1330-4b49-beda-2673249e625d";

        string res =
controller.UpdateUjianTest(id_guru,
id_ujian, Nama_Ujian, Enrolment_Key,
Mulai_Ujian, Batas_Ujian,
Durasi_pengerjaan);

        Assert.AreEqual("SUCCESS",
res);
    }

    [TestMethod]
    public void DeleteUjianA()
    {
        string id_guru = "61210bfb-
d1c3-41b2-8fe4-4c11033ab97a";

        string res =
controller.DeleteUjianTest(id_guru,
id_ujian);

        Assert.AreEqual("INVALID_CREATOR",
res);
    }

    [TestMethod]

```

```

        public void DeleteUjianB()
        {
            string id_guru = "36b83bf0-
1330-4b49-beda-2673249e625d";

            string res =
controller.DeleteUjianTest(id_guru,
id_ujian);

            Assert.AreEqual("SUCCESS",
res);
        }
    }
}

```

#### b. DetailUjianController.cs

```

using Microsoft.AspNet.Identity;
using
Microsoft.VisualStudio.TestTools.UnitTesting;
using SkripsiTagBasedExam.Models;
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.Entity;
using System.Linq;
using System.Net;
using System.Security.Principal;
using System.Web;
using System.Web.Mvc;
using UjianData.Detail;

namespace UjianData.Detail
{
    public class DetailUjian
    {
        public Ujian current { get;
set; }
        public List<EventUjian> events
{ get; set; }
        public List<Soal> soals { get;
set; }
    }
}

```

```

        public string Status { get;
set; }

        public DetailUjian()
        {
            current = new Ujian();
            events = new
List<EventUjian>();
            soals = new List<Soal>();
            Status = "NOT SUCCESS";
        }
    }

    public class Peserta
    {
        public Ujian current { get;
set; }
        public List<PesertaUjian>
list_peserta { get; set; }
        public List<DataSiswa> siswa {
get; set; }

        public Peserta()
        {
            current = new Ujian();
            list_peserta = new
List<PesertaUjian>();
            siswa = new
List<DataSiswa>();
        }
    }

    namespace
SkripsiTagBasedExam.Controllers
    {
        public class DetailUjianController
: Controller
        {
            private examdbEntities db = new
examdbEntities();

            [Authorize(Roles = "Guru")]

```

```

        public ActionResult Index(int?
id)
    {
        string id_guru =
User.Identity.GetUserId();

        if (id == null)
        {
            return
RedirectToAction("Index", "Ujian");
        }
        else
        {
            DetailUjian detail =
StoreLists(id_guru, (int)id);

            if(detail.Status ==
"SUCCESS")
            {
                ViewData["Guru"] =
db.DataGuru.Find(id_guru);

                return
View(detail);
            }
            else
            {
                return
RedirectToAction("Index", "Ujian");
            }
        }
    }

    //TEST PURPOSE
    public DetailUjian
StoreLists(string id_guru, int id)
    {
        try
        {
            DetailUjian dets = new
DetailUjian();
            var selected =
db.Ujian.Where(e => e.Ujian_Id == id);

```

```

            dets.current =
selected.ToList()[0];

            if(dets.current.Createdby == id_guru)
            {
                dets.events =
db.EventUjian.Include(e =>
e.Soal).Include(e => e.Ujian).Where(e
=> e.Ujian_Id == id).OrderBy(e =>
e.No).ToList();

                dets.soals =
db.Soal.Include(e => e.Tag).Where(e =>
e.Tag.MP_Id ==
dets.current.MP_Id).ToList();

                if
(dets.events.Count > 0)
                {
                    for (int i = 0;
i < dets.events.Count; i++)
                    {
                        for (int j
= 0; j < dets.soals.Count; j++)
                        {
                            if
(dets.soals[j].Soal_Id ==
dets.events[i].Soal_Id)
                            {
                                dets.soals.RemoveAt(j);
                            }
                        }
                    }
                }

                dets.Status =
"SUCCESS";
            }
            else
            {
                dets.Status =
"INVALID_CREATOR";
            }
        }
    }

```

```

        return dets;
    }
    catch
    {
        return new
DetailUjian();
    }
}

[Authorize(Roles = "Guru")]
public ActionResult
Peserta(int? id_ujian)
{
    string id_guru =
User.Identity.GetUserId();

    if (id_ujian == null ||
db.Ujian.Find(id_ujian).CreatedBy !=
id_guru)
    {
        return
RedirectToAction("Index", "Ujian");
    }
    else
    {
        Peserta peserta = new
Peserta();
        var selected =
db.Ujian.Where(e => e.Ujian_Id ==
id_ujian);
        peserta.current =
selected.ToList()[0];
        peserta.list_peserta =
db.PesertaUjian.Include(e =>
e.Ujian).Include(e =>
e.DataSiswa).Where(e => e.Ujian_Id ==
id_ujian).ToList();
        peserta.siswa =
db.DataSiswa.ToList();

        ViewData["Guru"] =
db.DataGuru.Find(id_guru);
        return View(peserta);
    }
}

```

```

    }

    [HttpPost]
    public string
searchSiswa(string q, int id_ujian)
    {
        string res = "";
        if (q != "")
        {
            var registered =
db.PesertaUjian.Where(e => e.Ujian_Id
== id_ujian).ToList();
            var siswas =
db.DataSiswa.Where(e =>
e>Nama_Siswa.Contains(q)).Take(5).ToLi
st();

            for (int i = 0; i <
registered.Count; i++)
            {
                for (int j = 0; j <
siswas.Count; j++)
                {
                    if
(siswas[j].Siswa_Id ==
registered[i].Siswa_Id)
                    {
                        siswas.RemoveAt(j);
                    }
                }
            }

            res += "<div
class='row'>";
            if (siswas.Count > 0)
            {
                for (int i = 0; i <
siswas.Count; i++)
                {
                    res += "<div
class='col-md-12'><span><button
onclick=\"add(" + id_ujian + ", '" +
siswas[i].Siswa_Id + "' );\" class='btn
btn-link'><span class='glyphicon

```



```

glyphicon-
plus'></span></button></span> " +
siswas[i].Nama_Siswa + "</div>";
        }
    }
    else
    {
        res += "<div
class='col-md-12'><p>Siswa tidak
ditemukan..</p></div>";
    }

    }

    return res;
}

[HttpPost]
public string
searchGrup(string q, int id_ujian)
{
    string res = "";
    if (q != "")
    {
        var registered =
db.PesertaUjian.Where(e => e.Ujian_Id
== id_ujian).ToList();

        var groups =
db.Grup.Where(e
=>
e>Nama_Grup.Contains(q)).Take(5).ToLis
t();

        for (int i = 0; i <
groups.Count; i++)
        {
            int grupid =
groups[i].Grup_Id;
            var members =
db.GrupMember.Where(e => e.Grup_Id ==
grupid).ToList();
            if (members.Count
> 0)
                {

```

```

                int
grupmembers = members.Count;
                int j = 0;

                for (int x = 0;
x < registered.Count; x++)
                {
                    for (int y
= 0; y < grupmembers; y++)
                    {
                        if
(registered[x].Siswa_Id ==
members[y].Siswa_Id)
                        {
                            j++;
                        }
                    }

                    if (j >=
grupmembers)
                    {
                        groups.RemoveAt(i);
                    }
                }

                res += "<div
class='row'>";
                if (groups.Count > 0)
                {
                    for (int i = 0; i <
groups.Count; i++)
                    {
                        res += "<div
class='col-md-12'><span><button
onclick=\"addgroup(\" + id_ujian + \", '\"
+ groups[i].Grup_Id + '\" );\" class='btn
btn-link'><span class='glyphicon
glyphicon-
plus'></span></button></span> " +
groups[i].Nama_Grup + "</div>";

```

```

        }
    }
    else
    {
        res += "<div
class='col-md-12'><p>Grup
ditemukan..</p></div>";
    }

    return res;
}

[Authorize(Roles = "Guru")]
public ActionResult
HapusSoal(int id_soal, int id_ujian)
{
    int i =
HapusSoalTest(id_soal);

    return
RedirectToAction("Index", new { id =
id_ujian });
}
//TEST PURPOSE
public int HapusSoalTest(int
id_soal)
{
    try
    {
        Soal _soal =
db.SoaI.Find(id_soal);
        db.SoaI.Remove(_soal);
        db.SaveChanges();
        return 1;
    }
    catch
    {
        return 0;
    }
}

[HttpPost]

```

```

    public int AddSoal(int
id_ujian, int id_soal)
    {
        int i = 0;
        try
        {
            int no =
db.EventUjian.Include(e =>
e.SoaI).Include(e => e.Ujian).Where(e
=> e.Ujian_Id ==
id_ujian).ToList().Count;

            EventUjian eventUjian
= new EventUjian();
            eventUjian.No = no + 1;
            eventUjian.Ujian_Id =
id_ujian;
            eventUjian.SoaI_Id =
id_soal;

            db.EventUjian.Add(eventUjian);
            db.SaveChanges();

            i = 1;
        }
        catch
        {
            i = 0;
        }
        return i;
    }

    [HttpPost]
    public int RemoveEvent(int
id_event, int id_ujian)
    {
        int i = 0;
        try
        {
            EventUjian eventUjian
= db.EventUjian.Find(id_event);

            int no =
(int)eventUjian.No;

```

```

db.EventUjian.Remove(eventUjian);
    db.SaveChanges();

    var    currentList    =
db.EventUjian.Where(e => e.Ujian_Id ==
id_ujian).ToList();
    for (int x = 0; x <
currentList.Count; x++)
    {
        if
(currentList[x].No > no)
        {
            no    =
(int)currentList[x].No;

currentList[x].No--;

db.Entry(currentList[x]).State    =
EntityState.Modified;

db.SaveChanges();

        }
    }

    i = 1;
}
catch
{
    i = 0;
}
return i;
}

[HttpPost]
public int SendUp(int id_event)
{
    int res = 0;

    EventUjian    current    =
db.EventUjian.Find(id_event);

    if (current.No > 1)

```

```

{
    var    checkprev    =
db.EventUjian.Where(e => e.No ==
current.No - 1).ToList();
    EventUjian    prev    =
checkprev[0];
    prev.No = current.No;
    db.Entry(prev).State =
EntityState.Modified;

    current.No    =
current.No - 1;

    db.Entry(current).State    =
EntityState.Modified;

    db.SaveChanges();
}
return res;
}

[HttpPost]
public int SendDown(int
id_event)
{
    int res = 0;

    EventUjian    current    =
db.EventUjian.Find(id_event);

    if (current.No <
db.EventUjian.Where(e => e.Ujian_Id ==
current.Ujian_Id).ToList().Count)
    {
        var    checkafter    =
db.EventUjian.Where(e => e.No ==
current.No + 1).ToList();
        EventUjian    prev    =
checkafter[0];
        prev.No = current.No;
        db.Entry(prev).State =
EntityState.Modified;

```

```

        current.No =
current.No + 1;

db.Entry(current).State =
EntityState.Modified;

        db.SaveChanges();
    }
    return res;
}

[HttpPost]
public string ViewSoal(int
id_soal)
{
    string res = "";

    Soal soal =
db.Soal.Find(id_soal);

    res += soal.Pertanyaan;

    res += "<br/><br/>";
    res += "A. " + soal.A +
"<br/>";
    res += "B. " + soal.B +
"<br/>";
    res += "C. " + soal.C +
"<br/>";
    res += "D. " + soal.D +
"<br/>";
    res += "E. " + soal.E +
"<br/><br/>";

    res += "Jawaban: " +
soal.Jawaban + "<br/>";
    res += "Tag: " +
soal.Tag.Tag1;

    return res;
}

[HttpPost]

```

```

public int AddCandidate(int
id_ujian, string id_siswa)
{
    int i = 0;
    try
    {
        var check =
db.PesertaUjian.Where(e => e.Siswa_Id
== id_siswa && e.Ujian_Id ==
id_ujian).ToList();

        if (check.Count < 1)
        {
            PesertaUjian
peserta = new PesertaUjian();
            peserta.Ujian_Id =
id_ujian;
            peserta.Siswa_Id =
id_siswa;
            peserta.Diambil =
0;

            db.PesertaUjian.Add(peserta);
            db.SaveChanges();
            i = 1;
        }
        else
        {
            i = 0;
        }
    }
    catch
    {
        i = 0;
    }
    return i;
}

[HttpPost]
public int AddGroup(int
id_ujian, int id_grup)
{
    int res = 0;

    try

```

```

        {
            var registered =
db.PesertaUjian.Where(e => e.Ujian_Id
== id_ujian).ToList();
            var members =
db.GrupMember.Where(e => e.Grup_Id ==
id_grup).ToList();

            for(int i = 0; i <
members.Count; i++)
            {
                bool exists =
false;
                for(int j = 0; j <
registered.Count; j++)
                {
                    if
(members[i].Siswa_Id ==
registered[j].Siswa_Id)
                    {
                        exists =
true;
                    }
                }
                if(exists ==
false)
                {
                    PesertaUjian
peserta = new PesertaUjian();
                    peserta.Ujian_Id = id_ujian;
                    peserta.Siswa_Id =
members[i].Siswa_Id;
                    peserta.Diambil = 0; // Belum di ambil
                    db.PesertaUjian.Add(peserta);
                    db.SaveChanges();
                    res = 1;
                }
            }
        }

```

```

        }
        catch
        {
            res = 0;
        }

        return res;
    }

    [HttpPost]
    public int RemoveCandidate(int
id_peserta)
    {
        int i = 0;
        try
        {
            int ID = id_peserta;
            PesertaUjian candidate
= db.PesertaUjian.Find(ID);
            db.PesertaUjian.Remove(candidate);
            db.SaveChanges();
            i = 1;
        }
        catch
        {
            i = 0;
        }
        return i;
    }

    protected override void
Dispose(bool disposing)
    {
        if (disposing)
        {
            db.Dispose();
        }
        base.Dispose(disposing);
    }
}

namespace UjianData.Tests
{

```

```

[TestClass]
public class DetailUjianTests
{
    GenericIdentity identity = new
GenericIdentity("Guru");

SkripsiTagBasedExam.Controllers.Detail
UjianController controller = new
SkripsiTagBasedExam.Controllers.Detail
UjianController();

    int id_ujian = 21;

    [TestMethod]
    public void IndexDetailA()
    {
        var id_guru = "61210bfb-
d1c3-41b2-8fe4-4c11033ab97a";
        var result =
controller.StoreLists(id_guru,
id_ujian);

Assert.AreEqual("INVALID_CREATOR",
result.Status);
    }

    [TestMethod]
    public void IndexDetailB()
    {
        var id_guru = "36b83bf0-
1330-4b49-beda-2673249e625d";
        var result =
controller.StoreLists(id_guru,
id_ujian);

        Assert.AreEqual("SUCCESS",
result.Status);
    }

    [TestMethod]
    public void AddSoalUjian()
    {
        var id_soal = 41;
        var result =
controller.AddSoal(id_ujian, id_soal);

```

```

        Assert.AreEqual(1,
result);
    }

    [TestMethod]
    public void RemoveSoalUjian()
    {
        var id_event = 97; //Insert
manually, check latest record in
EventUjian
        var result =
controller.RemoveEvent(id_event,
id_ujian);

        Assert.AreEqual(1,
result);
    }

    [TestMethod]
    public void AddPeserta()
    {
        var id_siswa = "63680f74-
2a73-492e-a151-896c959f1554";
        var result =
controller.AddCandidate(id_ujian,
id_siswa);

        Assert.AreEqual(1,
result);
    }

    [TestMethod]
    public void AddGroup()
    {
        var id_group = 1;
        var result =
controller.AddGroup(id_ujian,
id_group);

        Assert.AreEqual(1,
result);
    }

    [TestMethod]
    public void RemovePeserta()
    {
        var id_peserta = 76;
//insert manually, check latest record
in PesertaUjian

```

```

        var result =
controller.RemoveCandidate(id_peserta)
;
        Assert.AreEqual(1,
result);
    }
}

```

### c. SoalController.cs

```

using
Microsoft.VisualStudio.TestTools.UnitTesting;
using SkripsiTagBasedExam.Models;
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Security.Principal;
using System.Web;
using System.Web.Mvc;

namespace
SkripsiTagBasedExam.Controllers
{
    public class SoalController :
Controller
    {
        private examdbEntities db = new
examdbEntities();

        [HttpGet]
        public ActionResult
Index(string author, int? mapel, int?
ujian_id)
        {
            if (mapel == null || author
== null ||
db.DataGuru.SingleOrDefault(e =>
e.Guru_Id == author).MP_Id != mapel)
            {
                return
RedirectToAction("Index", "Ujian");

```

```

    }
    else
    {
        if (ujian_id != null)
        {
            ViewBag.Ujian =
ujian_id;
        }
        ViewData["Guru"] =
db.DataGuru.Find(author);
        ViewData["Mapel"] =
db.MataPelajaran.Find(mapel);
        ViewData["Tags"] =
db.Tag.Where(e => e.MP_Id ==
mapel).ToList();
        return View();
    }
}

[HttpPost]
public int Insert(string judul,
string author, int tag_id, string
question, string a, string b, string c,
string d, string e, string answer)
{
    int res = 0;

    try
    {
        Soal _new = new Soal();
        _new.Createdby =
author;
        _new.Tag_Id = tag_id;
        _new.Judul = judul;
        _new.Pertanyaan =
question;
        _new.A = a;
        _new.B = b;
        _new.C = c;
        _new.D = d;
        _new.E = e;
        _new.Jawaban = answer;
        _new.Tanggal =
DateTime.Now;

```

```

        db.SoaI.Add(_new);
        db.SaveChanges();
        res = 1;
    }
    catch
    {
        res = 0;
    }

    return res;
}
}
namespace UjianData.Tests
{
    [TestClass]
    public class SoalTests
    {
        GenericIdentity identity = new
        GenericIdentity("Guru");

        SkripsiTagBasedExam.Controllers.SoaICont
        ntroller controllerSoal = new
        SkripsiTagBasedExam.Controllers.SoaICont
        ntroller();

        SkripsiTagBasedExam.Controllers.Detail
        UjianController controllerDetail = new
        SkripsiTagBasedExam.Controllers.Detail
        UjianController();

        [TestMethod]
        public void CreateSoal()
        {
            string judul = "TEST_SOAL["
+
            DateTime.Now.ToString("yyyyMMddHHmmss"
            ) + "]";
            string author = "36b83bf0-
            1330-4b49-beda-2673249e625d";
            int tag_id = 3;
            string question =
            "TEST_QUESTION..";

```

```

        string a = "TEST ANSWER A";
        string b = "TEST ANSWER B";
        string c = "TEST ANSWER C";
        string d = "TEST ANSWER D";
        string e = "TEST ANSWER E";
        string answer = "A";

        var result =
        controllerSoal.Insert(judul, author,
        tag_id, question, a, b, c, d, e,
        answer);
        Assert.AreEqual(1,
        result);
    }

    [TestMethod]
    public void DeleteSoal()
    {
        int id_soal = 42;
        var result =
        controllerDetail.HapusSoalTest(id_soal
        );
        Assert.AreEqual(1,
        result);
    }
}
}

```



## d. UjianSiswaController.cs

```

using Microsoft.AspNet.Identity;
using
Microsoft.VisualStudio.TestTools.UnitTesting;
using SkripsiTagBasedExam.Models;
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.Entity;
using System.Data.Entity.Core.Objects;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.Mvc;

namespace
SkripsiTagBasedExam.Controllers
{
    public class UjianSiswaController :
Controller
    {
        private examdbEntities db = new
examdbEntities();

        [Authorize(Roles = "Siswa")]
        public ActionResult Index()
        {
            string id_siswa =
User.Identity.GetUserId();
            ViewData["Siswa"] =
db.DataSiswa.Find(id_siswa);
            ViewBag.SiswaID =
id_siswa;

            if (TempData["Message"] !=
null)
                ViewBag.Message =
TempData["Message"].ToString();

```

```

        UjianData.Status.UjianData
data = Inits(id_siswa);

        if
(db.OnGoingExam.ToList().Any(e =>
e.Siswa_Id == id_siswa))
        {
            return
RedirectToAction("Index",
"Clipboard");
        }
        else
        {
            return View(data);
        }
    }
    //TEST PURPOSE
    public
UjianData.Status.UjianData
Inits(string id_siswa)
    {
        try
        {
            DateTime today =
DateTime.Now;

            UjianData.Status.UjianData data = new
UjianData.Status.UjianData();

            data.done = new
List<Ujian>();
            var passed =
db.PesertaUjian.Where(e => e.Siswa_Id
== id_siswa && e.Diambil ==
1).ToList();

            for (int i = 0; i <
passed.Count; i++)
            {
                data.done.Add(db.Ujian.Find(passed[i].
Ujian_Id));
            }

            data.ongoing =
db.Ujian.Include(u =>

```

```

u.DataGuru).Include(u =>
u.MataPelajaran).Where(u =>
u.Mulai_Ujian <= today && u.Batas_ujian
> today).ToList();

        data.undone =
db.Ujian.Include(u =>
u.DataGuru).Include(u =>
u.MataPelajaran).Where(u =>
u.Mulai_Ujian > today).ToList();

        for (int i = 0; i <
data.undone.Count; i++)
        {
            var id =
data.undone[i].Ujian_Id;
            var peserta =
db.PesertaUjian.Where(e => e.Ujian_Id
== id).ToList();
            if (peserta.Any(e
=> e.Siswa_Id == id_siswa))
            {
                var get =
peserta.SingleOrDefault(e =>
e.Siswa_Id == id_siswa);
                if
(get.Diambil == 1)
                {
                    data.undone.RemoveAt(i);
                }
            }
        }

        for (int i = 0; i <
data.ongoing.Count; i++)
        {
            var id =
data.ongoing[i].Ujian_Id;
            var peserta =
db.PesertaUjian.Where(e => e.Ujian_Id
== id).ToList();
            if (peserta.Any(e
=> e.Siswa_Id == id_siswa))
            {

```

```

                var get =
peserta.SingleOrDefault(e =>
e.Siswa_Id == id_siswa);
                if
(get.Diambil == 1)
                {
                    data.ongoing.RemoveAt(i);
                }
            }
        }

        data.Status =
"SUCCESS";

        return data;
    }
    catch
    {
        return new
UjianData.Status.UjianData();
    }
}

public ActionResult
Participate(int id_ujian, string key)
{
    try
    {
        string id_siswa =
User.Identity.GetUserId();
        string message = "";

        switch
(ParticipateTest(id_siswa, key,
id_ujian))
        {
            case "SUCCESS":
                message =
"<div class='alert alert-success alert-
dismissable'><a href = '#'
class='close' data-dismiss='alert'
aria-label='close'>&times;</a>" +
                "Anda telah terdaftar
sebagai peserta ujian " +

```

```

db.Ujian.Find(id_ujian).Nama_Ujian +
". </div>";
                break;
            case
"INVALID_KEY":
                message =
"<div class='alert alert-danger alert-
dismissable'><a href = '#'
class='close' data-dismiss='alert'
aria-label='close'>&times;</a>" +
                "Kunci yang
anda masukan salah. </div>";
                break;
            }
            TempData["Message"] =
message;
            return
RedirectToAction("Index");
        }
        catch
        {
            return
RedirectToAction("Index");
        }
    }

    public string
ParticipateTest(string id_siswa,
string key, int id_ujian)
    {
        try
        {
            if (key ==
db.Ujian.Find(id_ujian).EnrolmentKey)
            {
                PesertaUjian pst =
new PesertaUjian();
                pst.Ujian_Id =
id_ujian;
                pst.Siswa_Id =
id_siswa;
                pst.Diambil = 0;

```

```

db.PesertaUjian.Add(pst);
                db.SaveChanges();
            }
            return "SUCCESS";
        }
        else
        {
            return
"INVALID_KEY";
        }
    }
    catch (Exception ex)
    {
        return ex.Message;
    }
}

    public ActionResult
StartExam(int id_ujian)
    {
        string id_siswa =
User.Identity.GetUserId();
        Ujian uji =
db.Ujian.Find(id_ujian);

        DateTime timeLeft =
DateTime.Now.AddMinutes(Double.Parse(u
ji.Durasi_pengerjaan.ToString()));

        Response.Cookies["Candidate"].Value =
id_siswa;

        OnGoingExam ongoing = new
OnGoingExam();
        ongoing.Ujian_Id =
uji.Ujian_Id;
        ongoing.FinishTime =
timeLeft;
        ongoing.Siswa_Id =
id_siswa;
        db.OnGoingExam.Add(ongoing);

```

```

        db.SaveChanges();

        return
RedirectToAction("Index", "Clipboard",
new { siswa = id_siswa });
    }

    [HttpPost]
    public string
PrepareWindow(int id_ujian)
    {
        string res = "";

        Ujian _for =
db.Ujian.Find(id_ujian);
        res += "Anda akan ikut
berpartisipasi dalam ujian: " +
_for>Nama_Ujian + "<br/><br/>";
        res += "Masukan enrolment
key untuk ujian ini ";
        res += "<form action='" +
Url.Action("Participate") + "'
method='POST'>";
        res += "<input
name='id_ujian' type='text'
class='collapse' value='" +
id_ujian.ToString() + "'><br/>";
        res += "<input name='key'
type='text' class='form-
control'><br/>";
        res += "<button class='btn
btn-primary'
type='submit'>Participasi</button></fo
rm>";

        return res;
    }

    [HttpPost]
    public string GetDetail(int
id_ujian)
    {
        string res = "";

```

```

        Ujian info =
db.Ujian.Find(id_ujian);

        res = "<p style='font-
size:24px; padding-top:10px;'>" +
info>Nama_Ujian + "</p>";
        res += "<div class='row'>";

        res += "<div class='col-md-
6'>";
        res += "<p>Mata Pelajaran
yang diuji: " +
info.MataPelajaran>Nama_MP + "<br/>";
        res += "Penyelenggara
ujian: " + info.DataGuru>Nama_Guru +
"<br/>";
        res += "Durasi pengerjaan:
" + info.Durasi_pengerjaan + "
Menit</p>";
        res += "</div>";

        res += "<div class='col-md-
6'>";
        res += "<p>Waktu mulai
ujian: " + info.Mulai_Ujian + "<br/>";
        res += "Batas waktu ambil
ujian: " + info.Batas_ujian + "</p>";
        res += "</div>";

        res += "</div>";

        return res;
    }
}
namespace UjianData.Tests
{
    [TestClass]
    public class UjianSiswaTests
    {

        SkripsiTagBasedExam.Controllers.UjianS
iswaController controller = new
SkripsiTagBasedExam.Controllers.UjianS
iswaController();

```

```

[TestMethod]
public void InitList()
{
    var id_siswa = "4c59f505-
fcee-4c4f-97d2-1630060d28cd";
    var result =
controller.Inits(id_siswa);
    Assert.AreEqual("SUCCESS",
result.Status);
}

[TestMethod]
public void ParticipateTestA()
{
    var id_siswa = "4c59f505-
fcee-4c4f-97d2-1630060d28cd";
    string key = "lol";
    int id_ujian = 21;

    var res =
controller.ParticipateTest(id_siswa,
key, id_ujian);
Assert.AreEqual("INVALID_KEY", res);
}

[TestMethod]
public void ParticipateTestB()
{
    var id_siswa = "4c59f505-
fcee-4c4f-97d2-1630060d28cd";
    string key = "Hello34";
    int id_ujian = 21;

    var res =
controller.ParticipateTest(id_siswa,
key, id_ujian);
    Assert.AreEqual("SUCCESS",
res);
}
}

```

### e. ClipboardController.cs

```

using Microsoft.AspNet.Identity;
using
Microsoft.VisualStudio.TestTools.UnitTesting;
using Newtonsoft.Json;
using SkripsiTagBasedExam.Models;
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.Entity;
using System.Linq;
using System.Net;
using System.Security.Principal;
using System.Web;
using System.Web.Mvc;
using UjianData.Status;

namespace
SkripsiTagBasedExam.Controllers
{
    public class ClipboardController :
Controller
    {

        private examdbEntities db = new
examdbEntities();

        OnGoingExam ongoing = null;

        [Authorize(Roles = "Siswa")]
        public ActionResult Index()
        {
            var ongoings =
db.OnGoingExam.ToList();

            string siswa =
User.Identity.GetUserId();

            if (ongoings.Any(e =>
e.Siswa_Id == siswa))
            {

```

```

        ongoing =
ongoings.SingleOrDefault(e =>
e.Siswa_Id == siswa);
        ViewBag.Message =
getSoal((int)ongoing.Ujian_Id);

        DateTime finish =
(DateTime)ongoing.FinishTime;
        string organizedDate =
finish.Month + "/" + finish.Day + "/" +
finish.Year + " " + finish.Hour + ":" +
finish.Minute + ":" + finish.Second;
        ViewBag.TimeLeft =
organizedDate;

        ViewBag.OnGoing =
db.Ujian.Find(ongoing.Ujian_Id).Ujian_
Id;

        ViewBag.Candidate =
ongoing.Siswa_Id;

        return View();
    }
    else
    {
        if
(Request.Cookies["Candidate"] != null)
        {

Response.Cookies["Candidate"].Expires
= DateTime.Now.AddDays(-1);
        }
        return
RedirectToAction("Index",
"UjianSiswa");
    }

    public ActionResult Finish()
    {
        string id =
User.Identity.GetUserId();
        var ongoings =
db.OnGoingExam.ToList();

```

```

        ongoing =
ongoings.SingleOrDefault(e =>
e.Siswa_Id == id);
        int ujian =
(int)ongoing.Ujian_Id;
        PesertaUjian peserta =
db.PesertaUjian.SingleOrDefault(u =>
u.Ujian_Id == ujian && u.Siswa_Id ==
id);

        peserta.Diambil = 1;
        List<Nilai> _result = new
List<Nilai>();
        string res =
SaveNilaiTest(id, ujian, _result);
        db.Entry(peserta).State =
EntityState.Modified;
        db.SaveChanges();

db.OnGoingExam.RemoveRange(db.OnGoingE
xam.Where(e => e.Ujian_Id == ujian &&
e.Siswa_Id == id));
        db.SaveChanges();

db.Clipboard.RemoveRange(db.Clipboard.
Where(e => e.Ujian_Id == ujian &&
e.Siswa_Id == id));
        db.SaveChanges();

        ViewData["Result"] =
_result;
        return View();
    }
    public List<Events>
getSoal(int id_ujian)
    {
        List<Events> res = new
List<Events>();
        var events =
db.EventUjian.Where(e => e.Ujian_Id ==
id_ujian).OrderBy(e => e.No).ToList();
        for (int i = 0; i <
events.Count(); i++)
        {
            res.Add(new Events(

```

```

        (int)events[i].No,
(int)events[i].Soal_Id,
(string)events[i].Soal.Pertanyaan,
(string)events[i].Soal.A,
(string)events[i].Soal.B,
(string)events[i].Soal.C,
(string)events[i].Soal.D,
(string)events[i].Soal.E
    ));
    }
    return res;
}

[HttpPost]
public string SaveChoices(int
no, string choice)
{
    string id =
User.Identity.GetUserId();
    return SaveChoicesTest(id,
no, choice);
}

[HttpPost]
public string GoTo(int no)
{
    string id =
User.Identity.GetUserId();
    string choice = "-";
    ongoing =
db.OnGoingExam.SingleOrDefault(e =>
e.Siswa_Id == id);

    int ujian =
(int)ongoing.Ujian_Id;
    string siswa =
ongoing.Siswa_Id;

```

```

        var clips =
db.Clipboard.Where(e => e.Ujian_Id ==
ujian && e.Siswa_Id == siswa).ToList();

        if (clips.Any(e => e.Nomor
== no))
        {
            for (int i = 0; i <
clips.Count; i++)
            {
                if (clips[i].Nomor
== no)
                {
                    choice =
clips[i].Choice;
                }
            }
        }
        string res = "";
        Events current =
getSoal(ujian)[no - 1];

        var evts =
db.EventUjian.Where(e => e.Ujian_Id ==
ujian).ToList();
        res += "Nomor <strong>" +
no + "</strong> dari <strong>" +
evts.Count + "</strong><br/><br/>";
        res += "<div>";
        res += current.Soal;
        res += "<br/>";
        res += "<div
class='radio'><label><input "; if
(choice == "A") res += "checked"; res
+= " type = 'radio' value = 'A'
onclick='saveCookie(" + no.ToString() +
");' name = 'Options'>" + current.A +
"</label></div>";
        res += "<div
class='radio'><label><input "; if
(choice == "B") res += "checked"; res
+= " type = 'radio' value = 'B'
onclick='saveCookie(" + no.ToString() +
");' name = 'Options'>" + current.B +
"</label></div>";

```

```

        res += "<div
class='radio'><label><input "; if
(choice == "C") res += "checked"; res
+= " type = 'radio' value = 'C'
onclick='saveCookie(" + no.ToString() +
");' name = 'Options'>" + current.C + "
</label></div>";
        res += "<div
class='radio'><label><input "; if
(choice == "D") res += "checked"; res
+= " type = 'radio' value = 'D'
onclick='saveCookie(" + no.ToString() +
");' name = 'Options'>" + current.D + "
</label></div>";
        res += "<div
class='radio'><label><input "; if
(choice == "E") res += "checked"; res
+= " type = 'radio' value = 'E'
onclick='saveCookie(" + no.ToString() +
");' name = 'Options'>" + current.E + "
</label></div>";
        res += "<hr/>";
        res += "</div>";
        res += "<button "; if (no
<= 1) res += "disabled"; res += "
onclick = 'GoTo(" + (no - 1) + ")'
class='btn ";
        if (no <= 1)
            res += "disabled";
        res += " btn-sm btn-
primary'>Before</button> ";
        res += "<button "; if (no
>= getSoal(ujian).Count) res +=
"disabled"; res += " onclick = 'GoTo("
+ (no + 1) + ")';' class='btn ";
        if (no >=
getSoal(ujian).Count)
            res += "disabled";
        res += " btn-sm btn-
primary'>Next</button> ";
        res += "<a
onclick='saveCookie(" + no.ToString() +
");' href='" + Url.Action("Finish") +
"'
class='btn btn-sm btn-
primary'>Finsih</a>";

```

```

        res += "<script> function
saveCookie(no){";
        res += "var selected =
$('input[name=\"Options\"]:checked').v
al();";
        res += " var events = {no:
no, choice: selected};";
        res += "$.ajax({";
        res += " url:
\"/Clipboard/SaveChoices\";";
        res += "data:
JSON.stringify(events),";
        res += "type: \"POST\";";
        res += "contentType:
\"application/json;charset=utf-8\";";
        res += "success:
function(result) {
console.log(result);";
        res += "}}});";
        res += "</script>";
        return res;
    }
    [HttpGet]
    public string GetReview()
    {
        string id =
User.Identity.GetUserId();
        string res = "";
        ongoing =
db.OnGoingExam.SingleOrDefault(e =>
e.Siswa_Id == id);
        int ujian =
(int)ongoing.Ujian_Id;
        string siswa =
ongoing.Siswa_Id;

        var clips =
db.Clipboard.Where(e => e.Ujian_Id ==
ujian && e.Siswa_Id == siswa).ToList();
        var evts =
db.EventUjian.Where(e => e.Ujian_Id ==
ujian).ToList();

        for (int i = 0; i <
evts.Count; i++)

```



```

        {
            res += "<button
class='btn btn-sm btn-';
                string style =
"default'";
                for (int j = 0; j <
clips.Count; j++)
                {
                    if (clips[j].Nomor
== evts[i].No)
                    {
                        string ans =
clips[j].Choice;
                        if (ans == "A"
|| ans == "B" || ans == "C" || ans ==
"D" || ans == "E")
                        {
                            style =
"success'";
                        }
                    }
                }
                res += style;
                res += " onclick =
'GoTo(" + (evts[i].No) + ")' data-
dismiss='modal'>" + evts[i].No;
                res += "</button>";
            }

            return res;
        }

//TEST PURPOSE
public string
SaveChoicesTest(string id, int no,
string choice)
{
    try
    {
        if (choice == null)
            choice = "-";
    }
}

```

```

        ongoing =
db.OnGoingExam.SingleOrDefault(e =>
e.Siswa_Id == id);

        var clips =
db.Clipboard.Where(e => e.Siswa_Id ==
id && e.Ujian_Id ==
ongoing.Ujian_Id).ToList();

        if (clips.Any(e =>
e.Nomor == no))
        {
            for (int i = 0; i <
clips.Count; i++)
            {
                if
(clips[i].Nomor == no)
                {
                    clips[i].Choice = choice;

                    db.Entry(clips[i]).State =
EntityState.Modified;
                }
            }
        }
        else
        {
            Clipboard clip =
new Clipboard();
            clip.Ujian_Id =
(int)ongoing.Ujian_Id;
            clip.Siswa_Id =
ongoing.Siswa_Id;
            clip.Nomor = no;
            clip.Choice =
choice;

            db.Clipboard.Add(clip);
        }

        db.SaveChanges();
        return "SUCCESS";
    }
}

```

```

        catch
        {
            return "NOT SUCCESS";
        }
    }

    public string
    SaveNilaiTest(string id, int ujian,
    List<Nilai> _result)
    {
        try
        {
            var questions =
            db.EventUjian.Where(e => e.Ujian_Id ==
            ujian).OrderBy(e => e.No).ToList();
            var inputs =
            db.Clipboard.Where(e => e.Ujian_Id ==
            ujian && e.Siswa_Id == id).OrderBy(e =>
            e.Nomor).ToList();
            for (int i = 0; i <
            questions.Count(); i++)
            {
                Nilai score = new
                Nilai();

                score.EventUjian_Id =
                questions[i].EventUjian_Id;
                score.Siswa_Id =
                id;

                string correct =
                questions[i].Soal.Jawaban;

                string jawaban = "-
                ";

                int nilai = 0;

                for (int j = 0; j <
                inputs.Count; j++)
                {
                    if
                    (inputs[j].Nomor == questions[i].No)
                    {

```

```

                    jawaban =
                    inputs[j].Choice;

                    if
                    (jawaban == questions[i].Soal.Jawaban)
                    nilai
                    = 1;
                }
            }

            score.Jawaban_Siswa = jawaban;
            score.Nilai1 =
            nilai;

            _result.Add(score);

            db.Nilai.Add(score);

            db.SaveChanges();
        }
        return "SUCCESS";
    }
    catch
    {
        return "NOT SUCCESS";
    }
}

class InputSiswa
{
    public int no { get; set; }
    public string choice { get;
    set; }

    public InputSiswa(int _no,
    string _choice)
    {
        no = _no; choice = _choice;
    }
}

```

```

}
namespace UjianData.Status
{
    public class Events
    {
        public int No { get; set; }
        public int Soal_ID { get; set; }
    }

    public string Soal { get; set; }
}

    public string A { get; set; }
    public string B { get; set; }
    public string C { get; set; }
    public string D { get; set; }
    public string E { get; set; }

    public Events(int _no, int
_soalid, string _soal, string _a,
string _b, string _c, string _d, string
_e)
    {
        No = _no;
        Soal_ID = _soalid;
        Soal = _soal;
        A = _a; B = _b; C = _c;
        D = _d; E = _e;
    }
}

namespace UjianData.Tests
{
    [TestClass]
    public class ClipboardTest
    {
        GenericIdentity identity = new
GenericIdentity("Guru");

        SkripsiTagBasedExam.Controllers.Clipbo
ardController controller = new
SkripsiTagBasedExam.Controllers.Clipbo
ardController();

```

```

        [Ignore]
        [TestMethod]
        public void GetSoal()
        {
            int id_ujian = 21;
            examdbEntities db = new
examdbEntities();
            var expected =
db.EventUjian.Where(e => e.Ujian_Id ==
id_ujian).ToList().Count();
            Assert.AreEqual(expected,
controller.getSoal(id_ujian).Count());
        }

        [TestMethod]
        public void saveChoice()
        {
            var id_siswa = "4c59f505-
fcee-4c4f-97d2-1630060d28cd";
            int no = 1;
            string choice = "A";

            var result =
controller.SaveChoicesTest(id_siswa,
no, choice);
            Assert.AreEqual("SUCCESS",
result);
        }

        [TestMethod]
        public void saveNilai()
        {
            string id = "4c59f505-fcee-
4c4f-97d2-1630060d28cd";
            int ujian = 21;
            List<Nilai> result = new
List<Nilai>();
            var res =
controller.SaveNilaiTest(id, ujian,
result);
            Assert.AreEqual("SUCCESS",
result);
        }
    }
}

```

## Source Code View Sistem

### a. Ujian/index.cshtml

```
@model UjianData.Status.UjianData

@{
    ViewBag.Title = "Daftar Ujian";
    Layout =
    "~/Views/Shared/_WithSidebar.cshtml";
}

@{
    SkripsiTagBasedExam.Models.DataGuru
    Guru =
    (SkripsiTagBasedExam.Models.DataGuru)V
    iewData["Guru"];
}

<link
href="cdn.rawgit.com/Eonasdan/bootstra
p-
datetimepicker/e8bddc60e73c1ec2475f827
be36e1957af72e2ea/build/css/bootstrap-
datetimepicker.css" rel="stylesheet">

<script
src="cdnjs.cloudflare.com/ajax/libs/mo
ment.js/2.9.0/moment-with-
locales.js"></script>
<script
src="cdn.rawgit.com/Eonasdan/bootstrap-
-
datetimepicker/e8bddc60e73c1ec2475f827
be36e1957af72e2ea/src/js/bootstrap-
datetimepicker.js"></script>

<div class="col-md-12">
    <div class="row">
        <div class="col-md-4">
            <a
href="@Url.Action("Details", "Guru")"
```

```
class="btn btn-link btn-sm"><span
class="glyphicon glyphicon-arrow-
left"></span> Kembali ke dashboard</a>
        </div>
    </div>
    <hr />
    <!--Body-->
    <button class="btn btn-sm btn-
primary" type="button" data-
toggle="modal" data-
target="#insert"><span
class="glyphicon glyphicon-plus-
sign"></span> Buat ujian baru</button>

    <br />
    <br />
    <div class="row">
        <div class="col-md-10">
            <p style="font-size:24px;
padding-top:10px;">Ujian yang akan
dilaksanakan</p>
            <div id="undone">
                @{
                    if
                    (Model.undone.Count > 0)
                    {
                        <table
class="table table-hover">
                            <tr>
                                <th
class="col-md-1">No.</th>
                                <th
class="col-md-5">Nama Ujian</th>
                                <th
class="col-md-5">Tanggal Mulai</th>
                                <th
class="col-md-1"></th>
                            </tr>
                            @{
                                int
                                counterUndone = 0;
                                foreach (var item in Model.undone)
                                    {
```

```

counterUndone++;

<tr>

<td>

@counterUndone

</td>

<td>

@Html.DisplayFor(modelItem =>
item>Nama_Ujian)

</td>

<td>

@Html.DisplayFor(modelItem =>
item>Mulai_Ujian)

</td>

<td>

<a class="btn btn-sm btn-primary"
href="@Url.Action("Index",
"DetailUjian", new { id = item.Ujian_Id
})">Lihat Detail</a>

</td>
</tr>

        }
    }
</table>
}
else
{

<p>Tidak ada data yang bisa di
perlihatkan..</p>

}

}

```

```

</div>

<hr />

<p style="font-size:24px;
padding-top:10px;">Ujian yang telah
dilaksanakan</p>

<div id="done">
    @{
        if
            (Model.done.Count > 0)
        {
            <table
class="table table-hover">
                <tr>
                    <th
class="col-md-1">No.</th>
                    <th
class="col-md-5">Nama Ujian</th>
                    <th
class="col-md-5">Tanggal Mulai</th>
                    <th
class="col-md-1"></th>
                </tr>
                @{
                    int
                    counterDone = 0;

                    foreach (var item in Model.done)
                        {
                            counterDone++;

                            <tr>

                            <td>

                            @counterDone

                            </td>

                            <td>

```

```

@Html.DisplayFor(modelItem =>
item>Nama_Ujian)

</td>

<td>

@Html.DisplayFor(modelItem =>
item>Mulai_Ujian)

</td>

<td>

<a class="btn btn-sm btn-primary"
href="@Url.Action("Index",
"DetailUjian", new { id = item.Ujian_Id
})">Lihat Detail</a>

</td>

</tr>

        }
    }
</table>
}
else
{
<p>Tidak ada data yang bisa di
perlihatkan..</p>
}
}
</div>
</div>
</div>

<div id="insert" class="modal fade"
role="dialog" tabindex="-1">
    <div class="modal-dialog">

        <!-- Modal content-->

```

```

        <div class="modal-content col-
md-9 col-lg-offset-1">
            <form id="insert-form"
method="post"
action="@Url.Action("InsertUjian",
"Ujian")">
                <div class="modal-
body">
                    <p style="font-
size:24px;"><span class="glyphicon
glyphicon-book"></span> Perbarui Data
Ujian</p>
                    <hr />
                    <div class="form-
group">
                        <p>Mata
Pelajaran:</p>
                        <input
disabled required name="MP_Id"
type="text" maxlength="20"
class="form-control" id="MP_Id"
value="@Guru.MataPelajaran>Nama_MP">
                        </div>
                        <div class="form-
group">
                            <p>Nama
Ujian:</p>
                            <input
required name="Nama_Ujian" type="text"
maxlength="20" class="form-control"
id="insert_nama" placeholder="Masukan
Nama Ujian">
                            </div>
                        <div class="form-
group">
                            <p>Kunci
Masuk:</p>
                            <input
required name="Enrolment_Key"
type="password" maxlength="10"
class="form-control" id="insert_key"
placeholder="Masukan Kata Kunci">
                            </div>

```

```

        <div class="form-
group">
            <p>Waktu Mulai
Ujian:</p>
            <input
required          name="Mulai_Ujian"
type="datetime"  class="form-control"
id="insert_start" placeholder="Masukan
Waktu Mulai">
            </div>
        <div class="form-
group">
            <p>Batas Ambil
Ujian:</p>
            <input
required          name="Batas_Ujian"
type="datetime"  class="form-control"
id="insert_end"  placeholder="Masukan
Batas Waktu">
            </div>
        <div class="form-
group">
            <p>Durasi
Pengerjaan:</p>
            <input
required          name="Durasi_pengerjaan"
type="number"     class="form-control"
id="insert_duration"
placeholder="Masukan          Durasi
Pengerjaan">
            </div>
        </div>
        <div class="modal-
footer">
            <button
type="submit"    class="btn    btn-
primary">Simpan Data Ujian</button>
            <button
type="button"   id="cancel"   class="btn

```

```

btn-default"      data-
dismiss="modal">Batal</button>
        </div>
    </form>
</div>

@section scripts{
    <script
src=~ /Scripts/moment.js"></script>
    <script    src=~ /Scripts/moment-
with-locales.js"></script>
    <script    src=~ /Scripts/bootstrap-
datetimepicker.min.js"></script>
    <link    href=~ /Content/bootstrap-
datetimepicker.min.css"
rel="stylesheet" />

    <script type="text/javascript">
        $(function () {
            var date = new Date();
            date.setDate(date.getDate());

            $('#insert_start').datetimepicker({
                locale: "id",
                format: "DD/MM/YYYY
HH:mm",
                minDate: date
            });

            $('#insert_end').datetimepicker({
                locale: "id",
                format: "DD/MM/YYYY
HH:mm",
                useCurrent: false
            });

            $('#insert_start').on("dp.change",
function (e) {

```

```

$('#insert_end').data("DateTimePicker")
).minDate(e.date);
    });

$('#insert_end').on("dp.change",
function (e) {

$('#insert_start').data("DateTimePicke
r").maxDate(e.date);
    });
    });
</script>
}

```

## b. DetailUjian/index.cshtml

```

@model UjianData.Detail.DetailUjian

@{
    ViewBag.Title = "Index";
    Layout =
    "~/Views/Shared/_WithSidebar.cshtml";
}

@{
    bool finished = false;

    if (Model.current.Mulai_Ujian <
DateTime.Now)
    {
        finished = true;
    }
}

<div class="row">
    <div class="col-md-4">
        <a href="@Url.Action("Index",
"Ujian" )" class="btn btn-link btn-
sm"><span class="glyphicon glyphicon-
arrow-left"></span> Kembali ke daftar
ujian</a>

```

```

</div>
</div>
<hr />
<p style="font-size:24px;"><span
class="glyphicon glyphicon-
check"></span>
@Model.current>Nama_Ujian </p>
<div class="row">
    <div class="col-md-11">
        <table class="table-
responsive" style="width:100%">
            <tr>
                <td>
                    <p>
                        Pelaksana
                    Ujian:
                    @Model.current.DataGuru>Nama_Guru<br
                    />
                        Kunci Masuk:
                    @Model.current.EnrolmentKey<br />
                        Tanggal
                    Dibuat: @Model.current.Tanggal_Upload
                    </p>
                </td>
                <td>
                    <p>
                        Tanggal Mulai:
                    @Model.current.Mulai_Ujian<br />
                        Batas Attempt:
                    @Model.current.Batas_ujian<br />
                        Durasi Ujian:
                    @Model.current.Durasi_pengerjaan Menit
                    </p>
                </td>
            </tr>
            <tr>
                <td colspan="2">
                    @if (finished ==
                    false)
                    {
                        <button
                        class="btn btn-sm btn-primary btn-
                        block" style="text-align: center"
                        type="button" data-toggle="modal"
                        data-target="#update"><span
                        class="glyphicon glyphicon-
                        edit"></span> Perbarui Ujian</button>

```



```

    }
    <button class="btn
btn-sm      btn-danger      btn-block"
style="text-align:      center"
type="button"      data-toggle="modal"
data-target="#delete"><span
class="glyphicon      glyphicon-
trash"></span> Hapus Ujian</button>
    </td>
  </tr>
</table>
</div>
</div>
<hr />
@{ string width = finished ? "col-md-
11" : "col-md-5"; }
<div class="row">
  <div class="@width">
    @if (finished == false)
    {
      <a class="btn btn-primary"
href="@Url.Action("Peserta",
"DetailUjian"      ,      new      {
id_ujian=Model.current.Ujian_Id
})"><span class="glyphicon glyphicon-
user"></span> Peserta Ujian</a>
    }
    <p      style="font-size:20px;
padding-top:10px;">Pertanyaan:</p>
    <div id="events">
      @if
      if (Model.events.Count
> 0)
      {
        <table
class="table table-hover">
          @foreach (var
item in Model.events)
          {
            <tr>
              <td>
                @Html.DisplayFor(modelItem =>
item.No).
              </td>

```

```

              <td>
                <button class="btn btn-link btn-sm"
type="button"      data-toggle="modal"
data-target="#viewsoalmodal"
onclick="viewSoal(@item.SoaI.SoaI_Id)"
>@Html.DisplayFor(modelItem =>
item.SoaI.Judul)</button>
              </td>
            @if
            (finished == false)
            {
              <td>
                @if ((int)item.No == 1)
                {
                  <button id="sendup" class="btn btn-sm
btn-link      no-space"
style="color:gray"><span
class="glyphicon      glyphicon-arrow-
up"></span></button>
                }
                else
                {
                  <button      id="sendup"
onclick="sendup(@item.EventUjian_Id)"
class="btn      btn-sm      btn-link      no-
space"><span      class="glyphicon
glyphicon-arrow-up"></span></button>
                }
                @if      ((int)item.No      ==
Model.events.Count)
                {
                  <button id="senddown" class="btn btn-sm
btn-link      no-space"

```

```

style="color:gray"><span
class="glyphicon glyphicon-arrow-
down"></span></button>
}
else
{
<button id="senddown"
onclick="senddown(@item.EventUjian_Id)
" class="btn btn-sm btn-link no-
space"><span class="glyphicon
glyphicon-arrow-down"></span></button>
}

<button id="remove"
onclick="remove(@item.EventUjian_Id,
@Model.current.Ujian_Id)" class="btn
btn-sm btn-link text-danger"><span
class="glyphicon glyphicon-
remove"></span></button>

</td>
}
</tr>
}
</table>
}
else
{
<p>Tidak ada data
yang bisa di perlihatkan..</p>
}
}
</div>
</div>
@if (finished == false)
{
<div class="col-md-6">

```

```

<a
href="@Url.Action("Index", "Soal" , new
{author =
Model.current.DataGuru.Guru_Id,
mapel=Model.current.MataPelajaran.MP_I
d, ujian_id=Model.current.Ujian_Id})"
class="btn btn-primary right"><span
class="glyphicon glyphicon-
plus"></span> Tambah Soal</a>

<p style="font-size:20px;
padding-top:10px;">Bank Soal:</p>
<div id="banksoal">

@{
if
(Model.soals.Count > 0)
{
<table
class="table table-hover">
@foreach
(var item in Model.soals)
{
<tr>

<td class="col-md-2">

<button id="tambah"
onclick="add(@Model.current.Ujian_Id,
@item.Soal_Id)" class="btn btn-link
btn-sm "><span class="glyphicon
glyphicon-arrow-left"></span>
Tambah</button>

</td>

<td class="col-md-6">

<button class="btn btn-link btn-sm"
type="button" data-toggle="modal"
data-target="#viewsoalmodal"
onclick="viewSoal(@item.Soal_Id)">@Htm

```

```

1.DisplayFor(modelItem =>
item.Judul)</button>

</td>

<td class="col-md-4">

<a class="btn btn-sm btn-link"><span
class="glyphicon glyphicon-
pencil"></span></a> <button
type="button" data-toggle="modal"
onclick="viewSoal(@item.SoaI_Id,1)"
data-target="#viewsoalmodal"
class="btn btn-sm btn-link text-
danger"><span class="glyphicon
glyphicon-trash"></span></button>

</td>

</tr>

}

</table>

}
else
{
<p>Tidak ada
data yang bisa di perlihatkan..</p>
}
}
</div>

</div>

}

</div>

<!-- Modal -->
<div id="delete" class="modal fade"
role="dialog">
<div class="modal-dialog">

<!-- Modal content-->
<div class="modal-content col-
md-8 col-lg-offset-2">
<div class="modal-body">

```

```

<p style="font-
size:24px;"><span class="glyphicon
glyphicon-trash"></span> Hapus Data
Ujian</p>

<hr />

<p>Anda akan menghapus
data @Model.current>Nama_Ujian. Data
yang telah dihapus tidak dapat
dikembalikan. Lanjutkan
penghapusan?</p>

</div>

<div class="modal-footer">
<button type="button"
onclick="deleteUjian(@Model.current.Uj
ian_Id)" class="btn btn-
danger">Lanjutkan</button>

<button type="button"
class="btn btn-default" data-
dismiss="modal">Batal</button>

</div>

</div>

</div>

<div>
<div id="update" class="modal fade"
role="dialog" tabindex="-1">
<div class="modal-dialog">

<!-- Modal content-->
<div class="modal-content
col-md-9 col-lg-offset-1">

<form id="update-form"
method="post"
action="@Url.Action("UpdateUjian",
"Ujian" )">

<div class="modal-
body">

<p
style="font-size:24px;"><span
class="glyphicon glyphicon-
book"></span> Perbarui Data Ujian</p>

<hr />

```

```

        <input
required          name="id_ujian"
value="@Model.current.Ujian_Id"
type="number" class="collapse">

        <div
class="form-group">
            <p>Nama
Ujian:</p>
            <input
required          name="Nama_Ujian"
value="@Model.current>Nama_Ujian"
type="text"       maxlength="20"
class="form-control" id="update_nama"
placeholder="Masukan Nama Ujian Baru">
            </div>
            <div
class="form-group">
            <p>Kunci
Masuk:</p>
            <input
required          name="Enrolment_Key"
value="@Model.current.EnrolmentKey"
type="text"       maxlength="10"
class="form-control" id="update_key"
placeholder="Masukan Kata Kunci Baru">
            </div>
            <div
class="form-group">
            <p>Waktu
Mulai Ujian:</p>
            <input
required          name="Mulai_Ujian"
value="@Model.current.Mulai_Ujian"
type="datetime" class="form-control"
id="update_start" placeholder="Masukan
Waktu Mulai">
            </div>
            <div
class="form-group">
            <p>Batas
Ambil Ujian:</p>
            <input
required          name="Batas_Ujian"
value="@Model.current.Batas_ujian"

```

```

type="datetime" class="form-control"
id="update_end" placeholder="Masukan
batas waktu">
        </div>
        <div
class="form-group">
            <p>Durasi
Pengerjaan:</p>
            <input
required          name="Durasi_pengerjaan"
value="@Model.current.Durasi_pengerjaa
n" type="number" class="form-control"
id="update_duration"
placeholder="Masukan          Durasi
Pengerjaan">
            </div>
        </div>
        <div class="modal-
footer">
            <button
type="submit" class="btn btn-
success">Perbarui Data</button>
            <button
type="button" class="btn btn-default"
data-dismiss="modal">Batal</button>
        </div>
    </form>
</div>
</div>
}
<!-- Modal -->
<div id="viewsoalmodal" class="modal
fade" role="dialog">
    <div class="modal-dialog">
        <!-- Modal content-->
        <div class="modal-content col-
md-10 col-lg-offset-1">
            <div class="modal-body">

```

```

        <p style="font-size:24px;"><span class="glyphicon glyphicon-list"></span> Detail Soal</p>
        <hr />
        <div id="soalviewer">
        </div>
    </div>
    <div class="modal-footer">
        <button type="button" class="btn btn-default" data-dismiss="modal">Tutup</button>
    </div>
</div>
</div>
<!-- Modal -->
<div id="deleteSoal" class="modal fade" role="dialog">
    <div class="modal-dialog">
        <!-- Modal content-->
        <div class="modal-content col-md-10 col-lg-offset-1">
            <div class="modal-body">
                <p style="font-size:24px;"><span class="glyphicon glyphicon-trash"></span> Hapus Soal</p>
                <hr />
                <div id="confirmdelete">
                </div>
            </div>
        </div>
    </div>

```

```

        </div>
    </div>
</div>
<script>
    function viewSoal(id,del) {
        var events = {
            id_soal: id
        };
        $.ajax({
            url:
                "/DetailUjian/ViewSoal",
            data:
                JSON.stringify(events),
            type: "POST",
            contentType:
                "application/json;charset=utf-8",
            success: function (result)
            {
                var content = result;
                if (del == 1) {
                    content +=
                        "<br/><br/><a href='@Url.Action(\"HapusSoal\", new { id_ujian = Model.current.Ujian_Id })&id_soal=" + id + "' class='btn btn-danger'><span class='glyphicon glyphicon-trash'></span> Hapus soal ini</a>";
                }
                $("#soalviewer").html(content)
            },
            error: function (errorMessage) {
                $("#soalviewer").html(errorMessage.responseText)
            }
        });
    }
</script>

```

```

function add(ujian, soal) {
    var events = {
        Id_ujian: ujian,
        Id_soal: soal
    };
    $.ajax({
        url:
"/DetailUjian/AddSoal",
        data:
JSON.stringify(events),
        type: "POST",
        contentType:
"application/json;charset=utf-8",
        dataType: "json",
        success: function (result)
{
    $('#events').load(document.URL + '
#events');

    $('#banksoal').load(document.URL + '
#banksoal');
        },
        error: function
(errorMessage) {
alert(errorMessage.responseText);
        }
    });
}

function remove(event, ujian) {
    var events = {
        id_event: event,
        id_ujian: ujian
    };
    $.ajax({
        url:
"/DetailUjian/RemoveEvent",
        data:
JSON.stringify(events),
        type: "POST",
        contentType:
"application/json;charset=utf-8",
        dataType: "json",

```

```

        success: function (result)
{
    $('#events').load(document.URL + '
#events');

    $('#banksoal').load(document.URL + '
#banksoal');
        },
        error: function
(errorMessage) {
alert(errorMessage.responseText);
        }
    });
}

function sendup(event) {
    var events = {
        id_event: event
    };
    $.ajax({
        url: "/DetailUjian/SendUp",
        data:
JSON.stringify(events),
        type: "POST",
        contentType:
"application/json;charset=utf-8",
        dataType: "json",
        success: function (result)
{
    $('#events').load(document.URL + '
#events');

    $('#banksoal').load(document.URL + '
#banksoal');
        },
        error: function
(errorMessage) {
alert(errorMessage.responseText);
        }
    });
}

```

```

function senddown(event) {
    var events = {
        id_event: event
    };
    $.ajax({
        url: "/DetailUjian/SendDown",
        data:
JSON.stringify(events),
        type: "POST",
        contentType:
"application/json;charset=utf-8",
        dataType: "json",
        success: function (result)
    {

$('#events').load(document.URL + '
#events');

$('#banksoal').load(document.URL + '
#banksoal');
    },
        error:          function
(errormessage) {
alert(errormessage.responseText);
    }
    });
}

function deleteUjian(ujian) {
    var events = {
        id_ujian: ujian
    };
    $.ajax({
        url: "/Ujian/DeleteUjian",
        data:
JSON.stringify(events),
        type: "POST",
        contentType:
"application/json;charset=utf-8",
        dataType: "json",

```

```

        success: function (result)
    {
        if (result === 1) {
            window.location.replace("@Url.Action("
Index", "Ujian")");
        }
    },
        error:          function
(errormessage) {
alert(errormessage.responseText);
    }
    });
}
}

</script>

@section scripts{
    <script
src="~/Scripts/moment.js"></script>
    <script src="~/Scripts/moment-
with-locales.js"></script>
    <script src="~/Scripts/bootstrap-
datetimepicker.min.js"></script>
    <link href="~/Content/bootstrap-
datetimepicker.min.css"
rel="stylesheet" />

    <script type="text/javascript">

        $(function () {
            var date = new Date();
            date.setDate(date.getDate());

$('#update_start').datetimepicker({
                locale: "id",
                format: "DD/MM/YYYY
HH:mm",
                minDate: date
            });

```

```

$('#update_end').datetimepicker({
    locale: "id",
    format: "DD/MM/YYYY
HH:mm",
    useCurrent: false
});

$("#update_start").on("dp.change",
function (e) {

$('#update_end').data("DateTimePicker"
).minDate(e.date);
});

$("#update_end").on("dp.change",
function (e) {

$('#update_start').data("DateTimePicke
r").maxDate(e.date);
});
}
</script>
}

```

```

</div>
</div>
<hr />
<p style="font-size:24px;"><span
class="glyphicon glyphicon-
user"></span> Peserta Ujian:
@Model.current>Nama_Ujian</p>

<div class="row">
    <div class="col-md-5">
        <br />
        <p>Siswa yang sudah terdaftar
menjadi peserta ujian:</p>

        <div id="candidates">
            <table class="table table-
hover">
                <tr>
                    <th>No.</th>
                    <th>
                        Nama Siswa
                    </th>
                    <th></th>
                </tr>
                @{ int i = 0;
                    foreach (var item in
Model.list_peserta)
                    {
                        i++;
                        <tr>
                            <td>
                                class="col-md-1">@i</td>
                            <td>
                                class="col-md-10">
                                    @Html.DisplayFor(modelItem =>
item.DataSiswa>Nama_Siswa)
                                </td>
                            <td>
                                class="col-md-1">
                                    <button
onclick="remove(@item.Peserta_Id)"
class="btn btn-link no-space"
style="color:red"><span

```

### c. DetailUjian/Peserta.cshtml

```

@model UjianData.Detail.Peserta
@{
    ViewBag.Title = "Peserta";
    Layout =
    "~/Views/Shared/_WithSidebar.cshtml";
}

<div class="row">
    <div class="col-md-4">
        <a href="@Url.Action("Index",
"DetailUjian", new { id =
Model.current.Ujian_Id })" class="btn
btn-link btn-sm"><span
class="glyphicon glyphicon-arrow-
left"></span> Kembali ke detail
ujian</a>

```



```

class="glyphicon glyphicon-
trash"></span></button>
</td>
</tr>
</table>
</div>
<div class="col-md-4">
<ul class="nav nav-tabs">
<li class="active"><a
data-toggle="tab"
href="#siswa">Siswa</a></li>
<li><a data-toggle="tab"
href="#grup">Grup</a></li>
</ul>
<div class="tab-content">
<div id="siswa"
class="tab-pane fade in active">
<br />
Masukkan nama siswa..
<input type="text"
id="namasiswa"
onkeyup="searchSiswa(@Model.current.Uj
ian_Id)" class="form-control" />
<br />
<div id="searchsiswa">
</div>
</div>
<div id="grup" class="tab-
pane fade">
<br/>
Masukkan nama grup..
<input type="text"
id="namagrup"
onkeyup="searchGrup(@Model.current.Uji
an_Id)" class="form-control" />
<br />
<div id="searchgrup">
</div>

```

```

</div>
</div>
</div>
@section scripts{
<script>
function add(ujian, siswa) {
var events = {
id_ujian: ujian,
id_siswa: siswa
};
$.ajax({
url:
"/DetailUjian/AddCandidate",
data:
JSON.stringify(events),
type: "POST",
contentType:
"application/json;charset=utf-8",
dataType: "json",
success: function
(result) {
// alert(result);
$('#candidates').load(document.URL + '
#candidates');
searchSiswa(ujian);
},
error: function
(errorMessage) {
alert(errorMessage.responseText);
}
});
function addgroup(ujian, grup)
{
var events = {
id_ujian: ujian,
id_grup: grup
};
$.ajax({

```

```

        url:
"/DetailUjian/AddGroup",
        data:
JSON.stringify(events),
        type: "POST",
        contentType:
"application/json;charset=utf-8",
        dataType: "json",
        success: function
(result) {
            //alert(result);

$('#candidates').load(document.URL + '
#candidates');
            searchGrup(ujian);
        },
        error: function
(errorMessage) {
            alert(errorMessage.responseText);
        }
    });
}
function searchSiswa(ujian) {
    var _q =
document.getElementById('namasiswa').v
alue;
    var events = {
        q: _q,
        id_ujian: ujian
    };
    $.ajax({
        url:
"/DetailUjian/searchSiswa",
        data:
JSON.stringify(events),
        type: "POST",
        contentType:
"application/json;charset=utf-8",
        success: function
(result) {
            // alert(result);

$('#searchsiswa').html(result);
        },

```

```

        error: function
(errorMessage) {
            alert(errorMessage.responseText);
        }
    });
}
function searchGrup(ujian) {
    var _q =
document.getElementById('namagrup').va
lue;
    var events = {
        q: _q,
        id_ujian: ujian
    };
    $.ajax({
        url:
"/DetailUjian/searchGrup",
        data:
JSON.stringify(events),
        type: "POST",
        contentType:
"application/json;charset=utf-8",
        success: function
(result) {
            // alert(result);

$('#searchgrup').html(result);
        },
        error: function
(errorMessage) {
            alert(errorMessage.responseText);
        }
    });
}
function remove(id) {
    var events = {
        id_peserta: id
    };
    $.ajax({
        url:
"/DetailUjian/RemoveCandidate",

```

```

        data:
JSON.stringify(events),
        type: "POST",
        contentType:
"application/json;charset=utf-8",
        dataType: "json",
        success: function
(result) {

$('#candidates').load(document.URL + '
#candidates');
        },
        error: function
(errorMessage) {
alert(errorMessage.responseText);
        }
    });
}
</script>
}

```

#### d. Soal/index.cshtml

```

@{
    ViewBag.Title = "Index";
    Layout =
"~/Views/Shared/_CustomLayout.cshtml";
}

@{
    string redirect =
"window.location.href = '' +
Url.Action("Index", "Ujian") + ''";
    string back = Url.Action("Index",
"Ujian");

    if (ViewBag.Ujian != null)
    {
        redirect =
"window.location.href = '' +

```

```

Url.Action("Index", "DetailUjian", new
{ id = ViewBag.Ujian }) + ''";
        back = Url.Action("Index",
"DetailUjian", new { id = ViewBag.Ujian
});
    }

SkripsiTagBasedExam.Models.DataGuru
Author =
(SkripsiTagBasedExam.Models.DataGuru)V
iewData["Guru"];

SkripsiTagBasedExam.Models.MataPelajar
an Subjects =
(SkripsiTagBasedExam.Models.MataPelaja
ran)ViewData["Mapel"];
}

<link
href="http://mindmup.github.io/bootstr
ap-wysiwyg/external/google-code-
prettify/prettify.css"
rel="stylesheet">
<script
src="http://mindmup.github.io/bootstra
p-
wysiwyg/external/jquery.hotkeys.js"></
script>
<script
src="http://mindmup.github.io/bootstra
p-wysiwyg/external/google-code-
prettify/prettify.js"></script>
<link
href="http://mindmup.github.io/bootstr
ap-wysiwyg/index.css"
rel="stylesheet">
<style>
    .choice {
        max-height: 100px;
        height: 100px;
        background-color: white;
        border-collapse: separate;
        border: 1px solid rgb(204, 204,
204);

```

```

padding: 4px;
box-sizing: content-box;
-webkit-box-shadow: rgba(0, 0, 0, 0.0745098) 0px 1px 1px 0px inset;
box-shadow: rgba(0, 0, 0, 0.0745098) 0px 1px 1px 0px inset;
border-top-right-radius: 3px;
border-bottom-right-radius: 3px;
border-bottom-left-radius: 3px;
border-top-left-radius: 3px;
overflow: scroll;
outline: none;
}
</style>

<br />
<div class="row">
  <div class="col-md-10 col-sm-offset-1">
    <p style="font-size:24px;">Soal baru untuk mata pelajaran @Subjects>Nama_MP</p>
    <hr />
    <div class="form-horizontal">
      <div class="form-group">
        <label class="col-sm-2 control-label">Pokok Bahasan</label>
        <div class="col-sm-6 selectContainer">
          <select class="form-control" id="tag">
            <option value="">Masukkan sebuah pokok bahasan</option>
            @foreach (var tag in ViewBag.Tags)
            <option value="@tag.Tag_Id">@tag.Tag1</option>
          </select>
        </div>
      </div>
    </div>
  </div>
</div>

```

```

<option value="@tags[i].Tag_Id">@tags[i].Tag1</option>
}
}
</select>
</div>
</div>
<div class="form-group">
  <label class="col-sm-2 control-label" for="judul">Judul Soal:</label>
  <div class="col-md-6">
    <input type="text" class="form-control" id="judul">
  </div>
</div>
</div>
<hr />
<div id="alerts"></div>
<div class="btn-toolbar" data-role="editor-toolbar" data-target="#editor">
  <div class="btn-group">
    <a class="btn btn-sm btn-primary dropdown-toggle" data-toggle="dropdown" title="Font Size"><span class="glyphicon glyphicon-text-size"></span>&nbsp;<b class="caret"></b></a>
    <ul class="dropdown-menu">
      <li><a data-edit="fontSize" style="font-size:5px;">Huge</a></li>
      <li><a data-edit="fontSize" style="font-size:3px;">Normal</a></li>
      <li><a data-edit="fontSize" style="font-size:1px;">Small</a></li>
    </ul>
  </div>

```

```

        <div class="btn-group">
            <a class="btn btn-sm
            btn-primary" data-edit="bold"
            title="Bold (Ctrl/Cmd+B)"><span
            class="glyphicon glyphicon-
            bold"></span></a>
            <a class="btn btn-sm
            btn-primary" data-edit="italic"
            title="Italic (Ctrl/Cmd+I)"><span
            class="glyphicon glyphicon-
            italic"></span></a>
            <a class="btn btn-sm
            btn-primary" data-edit="strikethrough"
            title="Strikethrough"><span
            style="text-decoration:line-
            through"><strong>S</strong></span></a>
            <a class="btn btn-sm
            btn-primary" data-edit="underline"
            title="Underline (Ctrl/Cmd+U)"><span
            style="text-
            decoration:underline"><strong>U</stron
            g></span></a>
        </div>
        <div class="btn-group">
            <a class="btn btn-sm
            btn-primary" data-
            edit="insertunorderedlist"
            title="Bullet list">List</a>
            <a class="btn btn-sm
            btn-primary" data-
            edit="insertorderedlist" title="Number
            list">Nomor</a>
        </div>
        <div class="btn-group">
            <a class="btn btn-sm
            btn-primary" data-edit="justifyleft"
            title="Align Left (Ctrl/Cmd+L)"><span
            class="glyphicon glyphicon-align-
            left"></span></a>
            <a class="btn btn-sm
            btn-primary" data-edit="justifycenter"
            title="Center (Ctrl/Cmd+E)"><span
            class="glyphicon glyphicon-align-
            center"></span></a>

```

```

            <a class="btn btn-sm
            btn-primary" data-edit="justifyright"
            title="Align Right (Ctrl/Cmd+R)"><span
            class="glyphicon glyphicon-align-
            right"></span></a>
            <a class="btn btn-sm
            btn-primary" data-edit="justifyfull"
            title="Justify (Ctrl/Cmd+J)"><span
            class="glyphicon glyphicon-align-
            justify"></span></a>
        </div>

        <div class="btn-group">
            <label class="btn btn-
            sm btn-primary">
                <span
                class="glyphicon glyphicon-
                picture"></span> <input data-
                role="magic-overlay" data-
                target="#pictureBtn" data-
                edit="insertImage" type="file"
                style="display: none;">
            </label>
        </div>
        <input type="text" data-
        edit="inserttext" id="voiceBtn" x-
        webkit-speech="">
    </div>
    <div id="editor">
        Pertanyaan..
    </div>
    <br /><br />

    <table class="table">
        <tr>
            <td class="col-md-1"
            valign="middle" align="center">
                <div
                class="radio">
                    <input
                    type="radio" value="A" name="choices">
                </div>
            </td>
            <td class="col-md-11">

```

```

        <div class="btn-
toolbar" data-role="A-toolbar" data-
target="#A">

                <div
class="btn-group">

                        <a
class="btn btn-sm btn-primary dropdown-
toggle" data-toggle="dropdown"
title="Font Size"><span
class="glyphicon glyphicon-text-
size"></span>&nbsp;<b
class="caret"></b></a>

                                <ul
class="dropdown-menu">

                                        <li><a
data-edit="fontSize 5"><font
size="5">Huge</font></a></li>

                                        <li><a
data-edit="fontSize 3"><font
size="3">Normal</font></a></li>

                                        <li><a
data-edit="fontSize 1"><font
size="1">Small</font></a></li>

                                </ul>

                        </div>

                <div
class="btn-group">

                        <a
class="btn btn-sm btn-primary" data-
edit="bold" title="Bold
( Ctrl/Cmd+B )" ><span class="glyphicon
glyphicon-bold"></span></a>

                                <a
class="btn btn-sm btn-primary" data-
edit="italic" title="Italic
( Ctrl/Cmd+I )" ><span class="glyphicon
glyphicon-italic"></span></a>

                                <a
class="btn btn-sm btn-primary" data-
edit="strikethrough"
title="Strikethrough"><span
style="text-decoration:line-
through"><strong>S</strong></span></a>

```

```

                <a
class="btn btn-sm btn-primary" data-
edit="underline" title="Underline
( Ctrl/Cmd+U )" ><span
style="text-
decoration:underline"><strong>U</stron
g></span></a>

                        </div>

                <div
class="btn-group">

                        <a
class="btn btn-sm btn-primary" data-
edit="insertunorderedlist"
title="Bullet list">List</a>

                                <a
class="btn btn-sm btn-primary" data-
edit="insertorderedlist" title="Number
list">Nomor</a>

                        </div>

                <div
class="btn-group">

                        <a
class="btn btn-sm btn-primary" data-
edit="justifyleft" title="Align Left
( Ctrl/Cmd+L )" ><span class="glyphicon
glyphicon-align-left"></span></a>

                                <a
class="btn btn-sm btn-primary" data-
edit="justifycenter" title="Center
( Ctrl/Cmd+E )" ><span class="glyphicon
glyphicon-align-center"></span></a>

                                <a
class="btn btn-sm btn-primary" data-
edit="justifyright" title="Align Right
( Ctrl/Cmd+R )" ><span class="glyphicon
glyphicon-align-right"></span></a>

                                <a
class="btn btn-sm btn-primary" data-
edit="justifyfull" title="Justify
( Ctrl/Cmd+J )" ><span class="glyphicon
glyphicon-align-justify"></span></a>

                        </div>

                <div
class="btn-group">

```

```

<label
class="btn btn-sm btn-primary">
  <span
class="glyphicon glyphicon-
picture"></span> <input data-
role="magic-overlay" data-
target="#pictureBtn" data-
edit="insertImage" type="file"
style="display: none;">
  </label>
</div>
<input
type="text" data-edit="inserttext"
id="voiceBtn" x-webkit-speech="">
</div>
<div id="A"
class="choice">
  Jawaban A..
</div>
</tr>
<tr>
<td class="col-md-1"
valign="middle" align="center">
  <div
class="radio">
    <input
type="radio" value="B" name="choices">
  </div>
</td>
<td class="col-md-11">
  <div class="btn-
toolbar" data-role="B-toolbar" data-
target="#B">
    <div
class="btn-group">
      <a
class="btn btn-sm btn-primary dropdown-
toggle" data-toggle="dropdown"
title="Font Size"><span
class="glyphicon glyphicon-text-
size"></span>&nbsp;<b
class="caret"></b></a>

```

```

<ul
class="dropdown-menu">
  <li><a
data-edit="fontSize 5"><font
size="5">Huge</font></a></li>
  <li><a
data-edit="fontSize 3"><font
size="3">Normal</font></a></li>
  <li><a
data-edit="fontSize 1"><font
size="1">Small</font></a></li>
</ul>
</div>
<div
class="btn-group">
  <a
class="btn btn-sm btn-primary" data-
edit="bold" title="Bold
( Ctrl/Cmd+B )" ><span class="glyphicon
glyphicon-bold"></span></a>
  <a
class="btn btn-sm btn-primary" data-
edit="italic" title="Italic
( Ctrl/Cmd+I )" ><span class="glyphicon
glyphicon-italic"></span></a>
  <a
class="btn btn-sm btn-primary" data-
edit="strikethrough"
title="Strikethrough"><span
style="text-decoration:line-
through"><strong>S</strong></span></a>
  <a
class="btn btn-sm btn-primary" data-
edit="underline" title="Underline
( Ctrl/Cmd+U )" ><span
style="text-
decoration:underline"><strong>U</stron
g></span></a>
</div>
<div
class="btn-group">
  <a
class="btn btn-sm btn-primary" data-
edit="insertunorderedlist"
title="Bullet list">List</a>

```

```

        <a
class="btn btn-sm btn-primary" data-
edit="insertorderedlist" title="Number
list">Nomor</a>
        </div>
        <div
class="btn-group">
            <a
class="btn btn-sm btn-primary" data-
edit="justifyleft" title="Align Left
(Ctrl/Cmd+L)"><span class="glyphicon
glyphicon-align-left"></span></a>
            <a
class="btn btn-sm btn-primary" data-
edit="justifycenter" title="Center
(Ctrl/Cmd+E)"><span class="glyphicon
glyphicon-align-center"></span></a>
            <a
class="btn btn-sm btn-primary" data-
edit="justifyright" title="Align Right
(Ctrl/Cmd+R)"><span class="glyphicon
glyphicon-align-right"></span></a>
            <a
class="btn btn-sm btn-primary" data-
edit="justifyfull" title="Justify
(Ctrl/Cmd+J)"><span class="glyphicon
glyphicon-align-justify"></span></a>
        </div>
        <div
class="btn-group">
            <label
class="btn btn-sm btn-primary">
                <span
class="glyphicon glyphicon-
picture"></span> <input data-
role="magic-overlay" data-
target="#pictureBtn" data-
edit="insertImage" type="file"
style="display: none;">
            </label>
        </div>

```

```

        <input
type="text" data-edit="inserttext"
id="voiceBtn" x-webkit-speech="">
        </div>
        <div id="B"
class="choice">
            Jawaban B..
        </div>
    </td>
</tr>
<tr>
    <td class="col-md-1"
valign="middle" align="center">
        <div
class="radio">
            <input
type="radio" value="C" name="choices">
        </div>
    </td>
    <td class="col-md-11">
        <div class="btn-
toolbar" data-role="C-toolbar" data-
target="#C">
            <div
class="btn-group">
                <a
class="btn btn-sm btn-primary dropdown-
toggle" data-toggle="dropdown"
title="Font Size"><span
class="glyphicon glyphicon-text-
size"></span>&nbsp;<b
class="caret"></b></a>
                <ul
class="dropdown-menu">
                    <li><a
data-edit="fontSize 5"><font
size="5">Huge</font></a></li>
                    <li><a
data-edit="fontSize 3"><font
size="3">Normal</font></a></li>
                    <li><a
data-edit="fontSize 1"><font
size="1">Small</font></a></li>
                </ul>

```



```

        </div>
        <div
class="btn-group">
            <a
class="btn btn-sm btn-primary" data-
edit="bold"          title="Bold
( Ctrl/Cmd+B )" ><span class="glyphicon
glyphicon-bold"></span></a>
            <a
class="btn btn-sm btn-primary" data-
edit="italic"        title="Italic
( Ctrl/Cmd+I )" ><span class="glyphicon
glyphicon-italic"></span></a>
            <a
class="btn btn-sm btn-primary" data-
edit="strikethrough"
title="Strikethrough"><span
style="text-decoration:line-
through"><strong>S</strong></span></a>
            <a
class="btn btn-sm btn-primary" data-
edit="underline"     title="Underline
( Ctrl/Cmd+U )" ><span style="text-
decoration:underline"><strong>U</stron
g></span></a>
        </div>
        <div
class="btn-group">
            <a
class="btn btn-sm btn-primary" data-
edit="insertunorderedlist"
title="Bullet list">List</a>
            <a
class="btn btn-sm btn-primary" data-
edit="insertorderedlist" title="Number
list">Nomor</a>
        </div>
        <div
class="btn-group">
            <a
class="btn btn-sm btn-primary" data-
edit="justifyleft"  title="Align Left
( Ctrl/Cmd+L )" ><span class="glyphicon
glyphicon-align-left"></span></a>

```

```

            <a
class="btn btn-sm btn-primary" data-
edit="justifycenter" title="Center
( Ctrl/Cmd+E )" ><span class="glyphicon
glyphicon-align-center"></span></a>
            <a
class="btn btn-sm btn-primary" data-
edit="justifyright" title="Align Right
( Ctrl/Cmd+R )" ><span class="glyphicon
glyphicon-align-right"></span></a>
            <a
class="btn btn-sm btn-primary" data-
edit="justifyfull"  title="Justify
( Ctrl/Cmd+J )" ><span class="glyphicon
glyphicon-align-justify"></span></a>
        </div>
        <div
class="btn-group">
            <label
class="btn btn-sm btn-primary">
                <span
class="glyphicon glyphicon-
picture"></span> <input data-
role="magic-overlay" data-
target="#pictureBtn" data-
edit="insertImage" type="file"
style="display: none;">
            </label>
        </div>
        <input
type="text" data-edit="inserttext"
id="voiceBtn" x-webkit-speech="">
        </div>
        <div id="C"
class="choice">
            Jawaban C..
        </div>
    </td>
</tr>
<tr>
    <td class="col-md-1"
valign="middle" align="center">

```

```

        <div
class="radio">
            <input
type="radio" value="D" name="choices">
        </div>
    </td>
    <td class="col-md-11">
        <div class="btn-
toolbar" data-role="D-toolbar" data-
target="#D">

            <div
class="btn-group">

                <a
class="btn btn-sm btn-primary dropdown-
toggle" data-toggle="dropdown"
title="Font Size"><span
class="glyphicon glyphicon-text-
size"></span>&nbsp;<b
class="caret"></b></a>

                <ul
class="dropdown-menu">

                    <li><a
data-edit="fontSize 5"><font
size="5">Huge</font></a></li>

                    <li><a
data-edit="fontSize 3"><font
size="3">Normal</font></a></li>

                    <li><a
data-edit="fontSize 1"><font
size="1">Small</font></a></li>

                </ul>

            </div>

            <div
class="btn-group">

                <a
class="btn btn-sm btn-primary" data-
edit="bold" title="Bold
( Ctrl/Cmd+B )" ><span class="glyphicon
glyphicon-bold"></span></a>

                <a
class="btn btn-sm btn-primary" data-
edit="italic" title="Italic
( Ctrl/Cmd+I )" ><span class="glyphicon
glyphicon-italic"></span></a>

```

```

        <a
class="btn btn-sm btn-primary" data-
edit="strikethrough"
title="Strikethrough"><span
style="text-decoration:line-
through"><strong>S</strong></span></a>

        <a
class="btn btn-sm btn-primary" data-
edit="underline" title="Underline
( Ctrl/Cmd+U )" ><span
style="text-
decoration:underline"><strong>U</stron
g></span></a>

    </div>

    <div
class="btn-group">

        <a
class="btn btn-sm btn-primary" data-
edit="insertunorderedlist"
title="Bullet list">List</a>

        <a
class="btn btn-sm btn-primary" data-
edit="insertorderedlist" title="Number
list">Nomor</a>

    </div>

    <div
class="btn-group">

        <a
class="btn btn-sm btn-primary" data-
edit="justifyleft" title="Align Left
( Ctrl/Cmd+L )" ><span class="glyphicon
glyphicon-align-left"></span></a>

        <a
class="btn btn-sm btn-primary" data-
edit="justifycenter" title="Center
( Ctrl/Cmd+E )" ><span class="glyphicon
glyphicon-align-center"></span></a>

        <a
class="btn btn-sm btn-primary" data-
edit="justiftright" title="Align Right
( Ctrl/Cmd+R )" ><span class="glyphicon
glyphicon-align-right"></span></a>

        <a
class="btn btn-sm btn-primary" data-
edit="justifyfull" title="Justify

```

```

(Ctrl/Cmd+J)"><span class="glyphicon
glyphicon-align-justify"></span></a>
</div>

<div
class="btn-group">
<label
class="btn btn-sm btn-primary">
<span
class="glyphicon glyphicon-
picture"></span> <input data-
role="magic-overlay" data-
target="#pictureBtn" data-
edit="insertImage" type="file"
style="display: none;">
</label>
</div>
<input
type="text" data-edit="inserttext"
id="voiceBtn" x-webkit-speech="">
</div>
<div id="D"
class="choice">
Jawaban D..
</div>
</td>
</tr>
<tr>
<td class="col-md-1"
valign="middle" align="center">
<div
class="radio">
<input
type="radio" value="E" name="choices">
</div>
</td>
<td class="col-md-11">
<div class="btn-
toolbar" data-role="E-toolbar" data-
target="#E">
<div
class="btn-group">

```

```

<a
class="btn btn-sm btn-primary dropdown-
toggle" data-toggle="dropdown"
title="Font Size"><span
class="glyphicon glyphicon-text-
size"></span>&nbsp;<b
class="caret"></b></a>
<ul
class="dropdown-menu">
<li><a
data-edit="fontSize 5"><font
size="5">Huge</font></a></li>
<li><a
data-edit="fontSize 3"><font
size="3">Normal</font></a></li>
<li><a
data-edit="fontSize 1"><font
size="1">Small</font></a></li>
</ul>
</div>
<div
class="btn-group">
<a
class="btn btn-sm btn-primary" data-
edit="bold" title="Bold
(Ctrl/Cmd+B)"><span class="glyphicon
glyphicon-bold"></span></a>
<a
class="btn btn-sm btn-primary" data-
edit="italic" title="Italic
(Ctrl/Cmd+I)"><span class="glyphicon
glyphicon-italic"></span></a>
<a
class="btn btn-sm btn-primary" data-
edit="strikethrough"
title="Strikethrough"><span
style="text-decoration:line-
through"><strong>S</strong></span></a>
<a
class="btn btn-sm btn-primary" data-
edit="underline" title="Underline
(Ctrl/Cmd+U)"><span style="text-
decoration:underline"><strong>U</stron
g></span></a>
</div>

```

```

<div
class="btn-group">
    <a
class="btn btn-sm btn-primary" data-
edit="insertunorderedlist"
title="Bullet list">List</a>
    <a
class="btn btn-sm btn-primary" data-
edit="insertorderedlist" title="Number
list">Nomor</a>
    </div>
<div
class="btn-group">
    <a
class="btn btn-sm btn-primary" data-
edit="justifyleft" title="Align Left
(Ctrl/Cmd+L)"><span class="glyphicon
glyphicon-align-left"></span></a>
    <a
class="btn btn-sm btn-primary" data-
edit="justifycenter" title="Center
(Ctrl/Cmd+E)"><span class="glyphicon
glyphicon-align-center"></span></a>
    <a
class="btn btn-sm btn-primary" data-
edit="justifyright" title="Align Right
(Ctrl/Cmd+R)"><span class="glyphicon
glyphicon-align-right"></span></a>
    <a
class="btn btn-sm btn-primary" data-
edit="justifyfull" title="Justify
(Ctrl/Cmd+J)"><span class="glyphicon
glyphicon-align-justify"></span></a>
    </div>
<div
class="btn-group">
    <label
class="btn btn-sm btn-primary">
    <span
class="glyphicon glyphicon-
picture"></span> <input data-
role="magic-overlay" data-
target="#pictureBtn" data-

```

```

edit="insertImage" type="file"
style="display: none;">
    </label>
</div>
<input
type="text" data-edit="inserttext"
id="voiceBtn" x-webkit-speech="">
</div>
<div id="E"
class="choice">
    Jawaban E..
</div>
</td>
</tr>
</table>

<div class="row">
    <div class="col-md-4 col-
md-offset-8">
    <a href="@back"
class="btn btn-warning btn-lg"><span
class="glyphicon glyphicon-arrow-
left"></span> Batal</a>
    <button class="btn
btn-lg btn-primary " id="save">
    <span
class="glyphicon glyphicon-
save"></span> Simpan Soal
    </button>
</div>
</div>
</div>

@section scripts{
    <script>
        $('#editor').wysiwyg();
        $('#A').wysiwyg({
toolbarSelector: '[data-role=A-
toolbar]' });
        $('#B').wysiwyg({
toolbarSelector: '[data-role=B-
toolbar]' });
    </script>

```

```

        $('#C').wysiwyg({
toolbarSelector: '[data-role=C-
toolbar]' });
        $('#D').wysiwyg({
toolbarSelector: '[data-role=D-
toolbar]' });
        $('#E').wysiwyg({
toolbarSelector: '[data-role=E-
toolbar]' });

        $('#save').click(function () {
            var _tag = $('#tag').val();
            var _judul =
$('#judul').val();
            var soal =
$('#editor').cleanHtml();
            var _a =
$('#A').cleanHtml();
            var _b =
$('#B').cleanHtml();
            var _c =
$('#C').cleanHtml();
            var _d =
$('#D').cleanHtml();
            var _e =
$('#E').cleanHtml();
            var selected =
$('#input[name="choices"]:checked').val
();

            if (_tag != "" && _judul !=
"" && soal != "" && _a != "" && _b !=
"" && _c != "" && _d != "" && _e != ""
&& selected != null) {

                var events = {
                    author:
'@Author.Guru_Id',
                    tag_id: _tag,
                    question: soal,
                    judul: _judul,
                    a: _a,
                    b: _b,
                    c: _c,
                    d: _d,

```

```

                    e: _e,
                    answer : selected
                };

                alert(JSON.stringify(events));
                $.ajax({
                    url:
"/Soal/Insert",
                    data:
JSON.stringify(events),
                    type: "POST",
                    contentType:
"application/json;charset=utf-8",
                    dataType: "json",
                    success: function
(result) {
                        if (result ==
1) {
                            @Html.Raw(redirect)
                        }
                    },
                    error: function
(errorMessage) {
                        alert(errorMessage.responseText);
                    }
                });
            }
            else {
                alert("Lengkap
data!");
            }
        });
    </script>
    <script>
        // We can attach the
`fileselect` event to all file inputs
on the page
        $(document).on('change',
':file', function () {
            var input = $(this),
                numFiles =
input.get(0).files
                ?
input.get(0).files.length : 1,

```

```

        label =
input.val().replace(/\\/g,
'/').replace(/.*\\//, '');

input.trigger('fileselect', [numFiles,
label]);
    });

    // We can watch for our custom
`fileselect` event like this
    $(document).ready(function ()
{
$(':file').on('fileselect', function
(event, numFiles, label) {

        var input =
$(this).parents('.input-
group').find(':text'),
        log = numFiles > 1
? numFiles + ' files selected' : label;

        if (input.length) {
            input.val(log);
        } else {
            if (log)
                alert(log);
        }

    });
});
</script>
}

```

#### e. UjianSiswa/index.cshtml

```

@model UjianData.Status.UjianData
@{
    ViewBag.Title = "Halaman Siswa";
    Layout =
"~/Views/Shared/_WithSidebar(Siswa).cs
html";
}

```

```

<div class="col-md-12">
    <div class="row">
        <div class="col-md-4">
            <a
href="@Url.Action("Details", "Siswa")"
class="btn btn-sm btn-link"><span
class="glyphicon glyphicon-arrow-
left"></span> Kembali ke dashboard</a>
        </div>
    </div>
    <hr />
    <!--Body-->
    @{
        if (ViewBag.Message != null)
        {
            <div>
                @Html.Raw(ViewBag.Message)
            </div>
        }
    }

    <div class="row">
        <div class="col-md-10">
            <p style="font-size:24px;
padding-top:10px;">Ujian yang sedang
dibuka</p>
            <div id="ongoing">
                @{
                    if
(Model.ongoing.Count > 0)
                    {
                        <table class="table
table-hover">
                            <tr>
                                <th
class="col-md-1">No.</th>
                                <th
class="col-md-3">Nama Ujian</th>
                                <th
class="col-md-3">Pelaksana Ujian</th>
                                <th
class="col-md-3">Tanggal Mulai</th>
                                <th
class="col-md-1"></th>
                            </tr>

```



```

                <th
class="col-md-3">Nama Ujian</th>
                <th
class="col-md-3">Pelaksana Ujian</th>
                <th
class="col-md-3">Tanggal Mulai</th>
                <th
class="col-md-1"></th>
            </tr>
            @{
                int no
= 0;
                foreach (var item in Model.undone)
                {
                    no++;
                    <tr>
                        <td>
                            @no
                        </td>
                        <td>
                            @Html.DisplayFor(modelItem
                            =>
                                item>Nama_Ujian)
                        </td>
                        <td>
                            @Html.DisplayFor(modelItem
                            =>
                                item>DataGuru>Nama_Guru)
                        </td>
                        <td>
                            @Html.DisplayFor(modelItem
                            =>
                                item>Mulai_Ujian)
                        </td>
                    </tr>
                }
            }

            if (item.PesertaUjian.ToList().Any(e
=> e.Siswa_Id == ViewBag.SiswaID))
            {

```

```

                <button data-toggle="modal" data-
target="#detail" class="btn btn-sm btn-
primary btn-block"
onclick="getDetail(@item.Ujian_Id)">Li
hat Detail</button>
            }
        }
        else
        {
            <button data-toggle="modal" data-
target="#participate" class="btn btn-
sm btn-warning btn-block"
onclick="prepareWindow(@item.Ujian_Id)
">Partisipasi</button>
        }
    }
}
</td>
</tr>
}
}
</table>
}
}
else
{
    <p>Tidak ada data yang
bisa di perlihatkan..</p>
}
}
}
</div>
<p style="font-size:24px;
padding-top:10px;">Ujian yang telah
dilaksanakan</p>

```





```

<div id="participate" class="modal
fade" role="dialog">
  <div class="modal-dialog">
    <!-- Modal content-->
    <div class="modal-content col-
md-10 col-lg-offset-1">
      <div class="modal-body">
        <p style="font-
size:24px;"><span class="glyphicon
glyphicon-list"></span>
Partisipasi</p>
        <hr />
        <div
id="participateWindow">
        </div>
      </div>
      <div class="modal-footer">
        <button type="button"
class="btn btn-default" data-
dismiss="modal">Tutup</button>
      </div>
    </div>
  </div>
<div id="detail" class="modal fade"
role="dialog">
  <div class="modal-dialog">
    <!-- Modal content-->
    <div class="modal-content col-
md-10 col-lg-offset-1">
      <div class="modal-body">
        <div id="detailujian">
        </div>
      </div>
      <div class="modal-footer">
        <button type="button"
class="btn btn-default" data-
dismiss="modal">Tutup</button>
      </div>
    </div>
  </div>
</div>
@section scripts{
  <script>

```

```

function prepareWindow(ujian)
{
  var events = {
    id_ujian: ujian
  };
  $.ajax({
    url:
"/UjianSiswa/PrepareWindow",
    data:
JSON.stringify(events),
    type: "POST",
    contentType:
"application/json;charset=utf-8",
    success: function
(result) {
      $("#participateWindow").html(result);
    },
    error: function
(errorMessage) {
      $("#participateWindow").html(errormess
age);
    }
  });
}
function getDetail(ujian) {
  var events = {
    id_ujian: ujian
  };
  $.ajax({
    url:
"/UjianSiswa/GetDetail",
    data:
JSON.stringify(events),
    type: "POST",
    contentType:
"application/json;charset=utf-8",
    success: function
(result) {
      $("#detailujian").html(result);
    },
    error: function
(errorMessage) {

```

```

$("#detailujian").html(errorMessage);
    }
    });
}
</script>
}

```

## f. Clipboard/index.cshtml

```

@{
    ViewBag.Title = "Index";
    Layout =
    "~/Views/Shared/_CustomLayout.cshtml";
}
<div class="row">
    <div class="col-md-8">
        <span class="glyphicon
glyphicon-list"></span> Ujian:
@ViewBag.OnGoing <span style="padding-
right:20px"></span> <span
class="glyphicon glyphicon-
user"></span> Peserta Ujian:
@ViewBag.Candidate
    </div>
    <div class="col-md-2">
        <button class="btn btn-sm btn-
primary" type="button" data-
toggle="modal" data-target="#review"
onclick="GetReview()"><span
class="glyphicon glyphicon-
tasks"></span> Tinjau Pekerjaan
Saya</button>
    </div>
    <div class="col-md-2">
        <p>Sisa Waktu: <span
id="timer"></span></p>
    </div>
</div>
<hr />
<div id="panel">
</div>
<!-- Modal -->

```

```

<div id="review" class="modal fade"
role="dialog">
    <div class="modal-dialog">
        <!-- Modal content-->
        <div class="modal-content col-
md-10 col-lg-offset-1">
            <div class="modal-body">
                <p style="font-
size:24px;"><span class="glyphicon
glyphicon-list"></span> Review
Ujian</p>
                <hr />
                <p>Keterangan: Hijau
adalah soal yang sudah dijawab dan
putih belum.</p>
                <br/>
                <div id="myreview">
                    </div>
            </div>
            <div class="modal-footer">
                <button type="button"
class="btn btn-default" data-
dismiss="modal">Tutup</button>
            </div>
        </div>
    </div>
</div>
@section scripts{
<script>
    function GoTo(no) {
        var events = {
            no: no
        };
        $.ajax({
            url: "/Clipboard/GoTo",
            data:
JSON.stringify(events),
            type: "POST",
            contentType:
"application/json;charset=utf-8",
            success: function (result)
        {

```

```

$("#panel").html(result);
    },
    error:          function
(errorMessage) {
$("#panel").html(errorMessage);
    }
});
}

function GetReview() {
$.ajax({
    url:
"/Clipboard/GetReview",
    type: "GET",
    contentType:
"application/json;charset=utf-8",
    success: function (result)
{
$("#myreview").html(result);
    },
    error:          function
(errorMessage) {
$("#myreview").html(errorMessage);
    }
});
}

</script>

<script>
    window.onload = function () {
        GoTo(1);
    };
    var    countDownDate    =    new
Date("@ViewBag.TimeLeft").getTime();
    // Set the date we're counting down
to
    var mins = 0;
    // Update the count down every 1
second
    var x = setInterval(function () {

```

```

        var initMins = false;

        // Get todays date and time
        var now = new Date().getTime();

        // Find the distance between
now an the count down date
        var distance = countDownDate -
now;

        // Time calculations for days,
hours, minutes and seconds
        var days = Math.floor(distance
/ (1000 * 60 * 60 * 24));
        var        hours        =
Math.floor((distance % (1000 * 60 * 60
* 24)) / (1000 * 60 * 60));
        var        minutes        =
Math.floor((distance % (1000 * 60 *
60)) / (1000 * 60));
        var        seconds        =
Math.floor((distance % (1000 * 60)) /
1000);

        // Display the result in the
element with id="demo"

document.getElementById("timer").inner
HTML = hours + ":"
        + minutes + ":" + seconds;

        // If the count down is
finished, write some text
        if (distance < 0) {
            window.location.href=
'@Url.Action("Finish")'
            clearInterval(x);

document.getElementById("timer").inner
HTML = "EXPIRED";
        }
    }, 1000);
</script>
}

```